SIMULATING THE KINETICS OF RARE BIOMOLECULAR EVENTS VIA OPTIMIZED FORWARD FLUX SAMPLING METHODS

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Ernesto Emmanuel Borrero

August 2009

SIMULATING THE KINETICS OF RARE BIOMOLECULAR EVENTS VIA OPTIMIZED FORWARD FLUX SAMPLING METHODS

Ernesto Emmanuel Borrero, Ph.D.

Cornell University 2009

Path sampling methods have become powerful simulation tools for understanding dynamics (i.e., kinetic constants and mechanism) in complex systems. However, the challenges of applying these path sampling algorithms, such as Forward Flux Sampling (FFS), to rare events include: (i) determination of an adequate order parameter; (ii) efficient partition and sampling of the phase space; and (iii) estimation of the free energy profile and barriers. The goals of the work were twofold: (1) to develop novel methods to enhance the efficiency of the simulation of rare-event transitions, and (2) to apply these methods to elucidate the kinetic mechanism of selected transitions in complex systems.

To address these shortcomings, an approach was first developed (FFS-LSE) for identifying suitable reaction coordinates to describe the progression of rare events in complex systems. A simple adaptive optimization algorithm was also proposed which optimizes the phase space sampling of the staging by concentrating the sampling in the "bottlenecks" of the FFS-type simulation. Finally, using a FFS-type algorithm as a framework to jump-start an Umbrella Sampling method (denoted FFS-US), it is shown that the kinetics and the free energy profile of the system can be evaluated in one single FFS simulation series. The applicability of these methods were illustrated by studying a biological switch and several energy surfaces having multiple metastable states, representative of complex kinetic behavior.

These approaches were also applied for the kinetic study of protein folding in the bulk and under confinement of a model lattice protein, offering a valuable insight

as to how the rapid initial formation of a critical core of amino acids (folding nuclei) affects the g lobal properties and the folding mechanism of a single chain protein. This study was then extended to the case of protein-fragment re-folding. It was found that the reassembly process of two model fragments is affected by the disruption of the "folding nucleus". Our results also show that the balanced distribution of the folding nucleus between protein fragments is key to their reassembly and a variable characterizing the formation of such nuclei (e.g. number of native contacts between fragments) should be included in the reaction coordinate model.

BIOGRAPHICAL SKETCH

Ernesto E. Borrero-Quintana studied in the Department of Chemical Engineering at University of Puerto Rico-Mayaguez Campus, where he earned a bachelor's degree (BS) in 2000 and master's degree (MS) in 2003. As an undergraduate student, he had the opportunity to collaborate in different researches oriented to environmental and renewable energy areas. After completing his BS, he continued his professional development and complemented it with a MS research thesis entitled "Optimization Studies for the alkaline Transesterification Biodiesel Reaction Using Ultrasound Mixing", in which he contributed to the key objective of developing a Center for Biodiesel Expertise in the Caribbean that could lead to and support a Biodiesel industry in Puerto Rico. In August 2004, he joined the Doctoral Program at Cornell University at Department of Chemical Engineering.

To my family

ACKNOLEDGEMENT

It has been a long and hard way to get to this point, the so expected day that I culminated my PhD studies. During this path, many people have contributed in numerous and distinct manners to the success of this professional goal. It is then fair to mention the most significant contributions, those who in determined moments, helped me to overcome the crucial steps of my academic and professional life. I am really grateful to Dr. Jose Colucci for his helpful counsel throughout my early professional growth. I would like to thank to my advisor, Prof. Fernando A. Escobedo, for his guidance throughout the project with his wonderful instruction and stimulating ideas. Thanks to him, this project, more than an opportunity to develop my research skills it was a pleasant educational experience. I also would like to thanks to Prof. Escobedo's research group and Lydia Contreras for their distinct ideas and suggestions that have contributed in numerous occasions to the polishing of the published work. To my family, I want to express my sincere love and gratitude for their presence, no matter the territorial miles that separated us for the past five years. Finally, I thank to my "*Reinita*", who supported me all the time and always believed in me.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. CHAPTER: Reaction coordinates and transition pathways of rare events via forward flux sampling*

Abstract

A new approach is developed for identifying suitable reaction coordinates to describe the progression of rare events in complex systems. The method is based on the forward flux sampling (FFS) technique and standard least square estimation (LSE) and it is denoted as FFS-LSE. The FFS algorithm generates trajectories for the transition between stable states as chains of partially connected paths, which can be then used to obtain "on-the-fly" estimates for the committor probability to the final region, p_B. These p_B data are then used to screen a set of candidate collective properties for an optimal order parameter (i.e., reaction coordinate) that depends on a few relevant variables. LSE is used to find the coefficients of the proposed reaction coordinate model and an analysis of variance is used to determine the significant terms in the model. The method is demonstrated for several test systems, including the folding of a lattice protein. It is shown that a simple approximation to p_B via a model linear on energy and number of native contacts is sufficient to describe the intrinsic dynamics of the protein system and to ensure an efficient sampling of pathways. In addition, since the p_B surface found from the FFS-LSE approach leads to the identification of the transition state ensemble, mechanistic details of the dynamics of the system can be readily obtained during a single FFS-type simulation without the need to perform additional committor simulations.

---

* E. E. Borrero and F. A. Escobedo, J. Chem. Phys. 127, 164101 (2007).

I. Introduction

For many processes in complex biological systems involving proteins, including folding, aggregation, adsorption, and transport-driven through membranes, the kinetics of such rare events is often very difficult to simulate by a conventional straightforward Molecular Dynamics (MD) approach.[1,2] This is because the average waiting time between events is orders of magnitude longer than the accessible simulation time itself, and most of the computational effort is spent simulating the uninteresting waiting period between events. Even when coarse-grained protein lattice models are simulated with dynamic Monte Carlo (MC), the required number of independent simulations needed to obtain accurate transition rate constants is extremely large.[3] To overcome this problem, specialized techniques have been proposed such as the Bennett-Chandler (BC) method, [4,5] effective positive flux (EPF) formalism,[6] the Crooks and Chandler approach,[7] transition path sampling (TPS),[8-11] milestoning methods,[12] transition interface sampling (TIS),[13,14] partial path transition interface sampling (PPTIS)[15] and "forward flux sampling" (FFS)-type simulation schemes.[2] These approaches can be classified as either "reactive flux" methods or "path sampling" methods. Reactive flux methods are based on a two-step approach wherein (i) the free energy profile as a function of a reaction coordinate is first determined from an initial simulation (e.g., via umbrella sampling or thermodynamic integration), and (ii) the transmission coefficient is calculated by firing dynamical trajectories from the top of the free energy barrier (~ the transition state diving surface) in a second independent simulation. There exist several techniques to determine the transmission coefficient such as the BC method,[16] the history dependent BC,[16] and the EPF formalism,[6] which differ in the way that the trajectories fired from the top energy barrier are counted. In these reactive flux methods, an improper choice of reaction coordinate will affect the correctness and efficiency of the transmission

coefficient calculation[17] (e.g., leading to very low coefficients). Moreover, these methods do not provide information about the transition path ensemble (TPE) or the transition states ensemble (TSE) to extract mechanistic details of the dynamics of the system.

Path sampling methods are based on an importance sampling of dynamical trajectories instead of the free energy barrier and transmission coefficient calculations. For example, TPS samples the TPE by using a MC procedure that generates new paths by "shooting" forward and backward in time from already existing paths of a fixed length in systems exhibiting stable states (i.e., equilibrium systems). The estimation of the average transition rate constants within the framework of TPS requires the definition of an order parameter, which is used only to distinguish the two stable states, and the calculation of a correlation function of state populations in time by an umbrella sampling approach.[7-9] Others methods, like the Crooks and Chandler approach, can be applied to non-equilibrium systems and rely on a methodology that resembles the TPS in generating new paths from old paths by changing the random number history.[7] In TIS[14] and PPTIS,[15] the efficiency of the TPE sampling is improved via a series of interfaces in phase space that facilitate the generation of flexible transition path lengths by a TPS-like procedure. Milestoning methods[12] are very similar to PPTIS, but trajectories between interfaces are generated assuming a steady-state distribution at each interface (i.e., a strong Markovian condition), an assumption that limits their applicability. Finally, FFS-type simulation algorithms exist that do not require the knowledge of the phase-space density and can therefore be used for non-equilibrium systems with stochastic dynamics. Like TIS, FFS allows the computation of both rate constants and TPE by dividing the phase space between the initial and final region into a series of interfaces. FFS relies on the generation of partial trajectories between interfaces, in which the crossing points at the next

interface of successful trial runs are stored and used to initiate new partial pathways for the following interface.[1,2,18] The TPE is obtained by connecting sequentially generated partial paths between the interfaces. The rate constant for the stochastic system is given by the product of the flux of trajectories crossing the first interface and the probability that these trajectories subsequently reach the final state. Although a series of interfaces in the phase-space is used as in other sampling techniques, no assumption is made about the distribution of paths at the interfaces;[2] i.e., trajectories are generated without assuming a steady-state distribution at each interface (or "memory loss" during transitions like in milestoning methods[12]) . Moreover, in FFS reactive trajectories are obtained without any requirement on their length as in TPS, creating more connected trajectories between the stable states than TIS with the same amount of MD steps, without relying on a Markovian assumption as in PPTIS.[17] These advantages of FFS made it the method of choice for this study.

From path sampling simulations one can obtain not only the transition path ensemble (TPE) between stables states but also the transition state ensemble (TSE), which is often very difficult to simulate by conventional "brute-force" approaches.[3,8] The TPE thus generated provides a viable method for obtaining mechanistic details and rate constants for the dynamics of the stochastic system. Moreover, all of these transition path methods avoid many of the difficulties associated with existing reactive flux methods and do not require the specification of a good reaction coordinate. However, it has been reported that the efficiency of these sampling schemes is sensitive to the choice of order parameter.[17] Hence, the challenges of applying a FFS-type algorithm to complex biological systems include: (i) determination of an adequate order parameter that allows the description of the transition state regions and (ii) assessing efficiency and completeness of sampling. In this paper, we attempt to address these two points by presenting a simple approach that identifies an adequate

reaction coordinate (i.e., order parameter) that leads to an efficient sampling in FFS-type simulations.

In high-dimensional complex systems, it is not a trivial task to find a good order parameter (i.e., a reaction coordinate) that quantifies progress along the reaction pathways and allows distinguishing the stable states of the system. The knowledge of the reaction coordinate is essential for the understanding of the dynamics of the rare event, and for the efficiency of the sampling algorithm.[17] The reaction coordinate is closely related to the probability of a configuration x to commit to the final state B; i.e., the "committor probability" $p_B(x)$,[19] which quantifies the tendency of configuration x to relax to the basin of attraction B under the system's intrinsic dynamics. Clearly, configurations in the initial basin A have $p_B = 0$, those in basin B have $p_B = 1$, and those at the TS have $p_B = ½$. Hence, p_B can be seen as a perfect reaction coordinate in the sense that it provides a quantitative description of the dynamic behavior of every state along a trajectory. But to be of practical use, p_B should be related to a few collective variables that are functions of the configurations and thus compress many atomistic details into physically important properties.[19]

In the conventional committor analysis,[11,19] a minimum number of fleeting trajectories, N_{min}, are initiated from a starting configuration along one of the paths belonging to the TPE and outside of the initial region A. The probability that a state in the phase-space committed to the final region is therefore estimated from the fraction of paths that end in state B. Unfortunately this procedure to estimate p_B is costly, requiring a huge computational effort because each p_B value requires on the order of $N_{min} = 10$ fleeting trajectories, each half as long as a reactive trajectory.[3,11] Furthermore, good statistics require hundreds of estimates for p_B histograms and analysis of ≥ 100 trajectories in the TPE.[19] The difficulties and computational cost of the committor analysis have motivated recent attempts to systematize the search for

reaction coordinates. However, the majority of these methods continue to use expensive histograms calculations,[20-24]and mainly focus on improving the trial and error aspects of the p_B calculation. For example, Ma and Dinner proposed a method based on neural networks to determine the functional dependence of p_B on a set of coordinates, and a genetic algorithm that selects the combination of inputs that yields the best fit via the estimation of p_B histograms.[22] Hummer introduced a new criterion for the TS as those points in configurational space with high probability p(TP|x) that equilibrium trajectories passing through them are reactive (i.e., connect stables states). The projection of p(TP|x) onto a good reaction coordinate should give a sharply peaked distribution, which can be used to choose among different candidate order parameters,[20,21] but it requires estimation of a p(TP|x) histogram for each iterative improvement of reaction coordinate model.[20,21] Weinan et al.[24] identify isocommittor surfaces (a costly procedure) to find effective transition tubes inside which the reactive trajectories stay confined (e.g., regions within the tubes where $p_B \sim ½$ define the TS). Maragliano et al.[23] combined a string method with a sampling technique to determine minimum free energy paths (MFEP). Their approach presumes that transitions are most likely to occur around the MFEP and thus isocommittor surfaces are determined therein. However, this approach requires many iterations of the mean force and variable entanglement calculations.[23]

A recently proposed approach for calculating reaction coordinates is based on TPS and likelihood maximization.[19] This so-called Maximum Likelihood (ML) method screens a set of candidate collective variables for a good reaction coordinate that depends on a few relevant variables. This is achieved by using a method denoted "aimless shooting" to harvest independent realizations of p_B. A simple model for the reaction coordinate; e.g., a lineal combination of the collective variables, is then assumed and used to calculate the likelihood of the model given the shooting data. The

Bayesian information criterion is used to determine significant variables for the reaction coordinate. At present, this ML method[19] and the one proposed by Maragliano et al.[23] are the only two approaches that do not rely on costly histograms.

Stimulated by the ML approach, we present in this article an alternative algorithm to obtain estimates for p_B "on-the-fly" from a FFS-type simulation. These p_B data are then used to fit a model for the reaction coordinate in terms of several collective variables. Standard least-squares estimation (LSE) is used to find the coefficients of the model and an analysis of variance (ANOVA) is used to determine the significant terms in the model. In this way, an optimized reaction coordinate that depends on a few relevant variables is obtained. Hereafter, we refer to this approach as the FFS-LSE method. Since the resulting model for the reaction coordinate corresponds to the p_B surface response, the TSE and hence the mechanistic details of the process can be readily obtained by only analyzing characteristics of the collective variables at the p_B contour of ½ $\pm \sigma$ (where σ is the desired level of statistical accuracy). Moreover, we show that the computational efficiency of the FFS method can be increased with the use of the optimized order parameter. We illustrate the application of the FFS-LSE method to several simple test systems, including the folding of a lattice protein model.

II. FFS-Type Simulations

By way of background, we start by briefly reviewing the formalism of FFS-type simulation schemes for calculating transition pathways; in particular, for the Branched Growth path sampling method.

The FFS-type algorithm estimates the rate constant of the process as an average rate of transitions from two well-defined states A and B using an "effective positive flux" expression.[1,2,13,14,18] To this end, it employs a series of nonintersecting

interfaces (n+1) between the initial (A) and final (B) regions $\{\lambda_0,\ldots\ldots,\lambda_n\}$ defined by one or a combination of order parameters (λ) , such that $\lambda \geq \lambda_0$, $\lambda_n = \lambda_B$, and $\lambda_i > \lambda_{i-1}$ (λ increases monotonically as the interfaces $\lambda_0,\ldots,\lambda_n$ are crossed). The order parameter $\lambda(x)$ can be chosen such that the system has values $\lambda(x) \leq \lambda_A(x)$ in region A and $\lambda(x) \geq \lambda_B(x)$ in region B. Here, x denotes the coordinates of the phase space. Hence, the rate constant $k_{A\to B}$ for transitions from A to B can be calculated from the total average flux from A to B, which can be expressed as the product of a flux from A to λ_0, $\overline{\Phi}_{A,0}$, and the probability that a trajectory reaching λ_0 from A will reach to B without returning to A, $P(\lambda_{n=B} \mid \lambda_0)$:[2,13,14]

$$k_{A\to B} = \frac{\overline{\Phi}_{A,0}}{\overline{h}_A} P(\lambda_{n=B} \mid \lambda_0) \qquad (1)$$

Here, the factor h_A is a history-dependent function such that $h_A = 1$ if the system was more recently in A than in B, and $h_A = 0$ otherwise; $\overline{h}_A$ is the average fraction of the time that the system spends in the basin of attraction of A. $P(\lambda_{n=B} \mid \lambda_0)$ can be expressed as the product of conditional probabilities:

$$P(\lambda_{n=B} \mid \lambda_0) = \prod_{i=0}^{n-1} P(\lambda_{i+1} \mid \lambda_i) \qquad (2)$$

where $P(\lambda_{i+1} \mid \lambda_i)$ is the probability that a trajectory that visits A and crosses λ_i for the first time will subsequently reach λ_{i+1} without returning to the initial region A.[2]

The rare paths (excursions or jumps between stable states in the free energy landscape) are captured by performing MC sampling of trajectories between interfaces. Furthermore, the paths belonging to the TPE are generated such that any trajectory from A to B passes through each interface in turn. The transitions between interfaces are free to follow any possible path between A and B, including paths

crossing previous interfaces several times. Three path sampling schemes have been proposed in the literature to generate transition paths including:[1,2] (i) the "Direct" Forward Flux Sampling (DFFS), (ii) Branched Growth method (BG), and (iii) Rosenbluth method (RB). Our FFS-LSE approach is based on the BG sampling scheme, which we implemented as described below.

The BG method is illustrated schematically in Figure 1.1(a), where branched transition paths are generated one by one. In the first stage of the algorithm, a simulation is carried out in the basin of attraction of A. After an equilibration period, $\lambda(x)$ is monitored during the run in the basin A and configurations crossing λ_0 are stored. This simulation is suspended when a large set of crossing points at λ_0 are generated (for a valid crossing the system must have visited A just before that) . In the second stage of the algorithm, a branched path is generated from a randomly chosen configuration at λ_0. This single configuration at λ_0 is used to initiate k_0 trial runs, which are continued until either reaching λ_1 or returning to the initial region. Each configuration ending in λ_1 is stored and used as starting point for k_1 trial runs to λ_2 or back to A. This procedure is repeated until either the final region $\lambda_n=\lambda_B$ is reached or because no successful trials were generated at some intermediate interfaces λ_i. In our implementation, the points saved at each λ_i were written on disk to avoid memory overload. An estimate of $P(\lambda_{n=B} \mid \lambda_0)$ is obtained as the total number of branches that eventually reach λ_n, divided by the total possible number of branches, i.e.,

$$P(\lambda_{n=B} \mid \lambda_0) = \frac{N_S^{(n-1)}}{\prod_{i=0}^{n-1} k_i} . \qquad (3)$$

A new branching path is then generated by randomly choosing another point at λ_0 and following the same procedure outlined above to get a new estimate of $P(\lambda_{n=B} \mid \lambda_0)$. The final estimate of $P(\lambda_{n=B} \mid \lambda_0)$ is then obtained from the average over all such

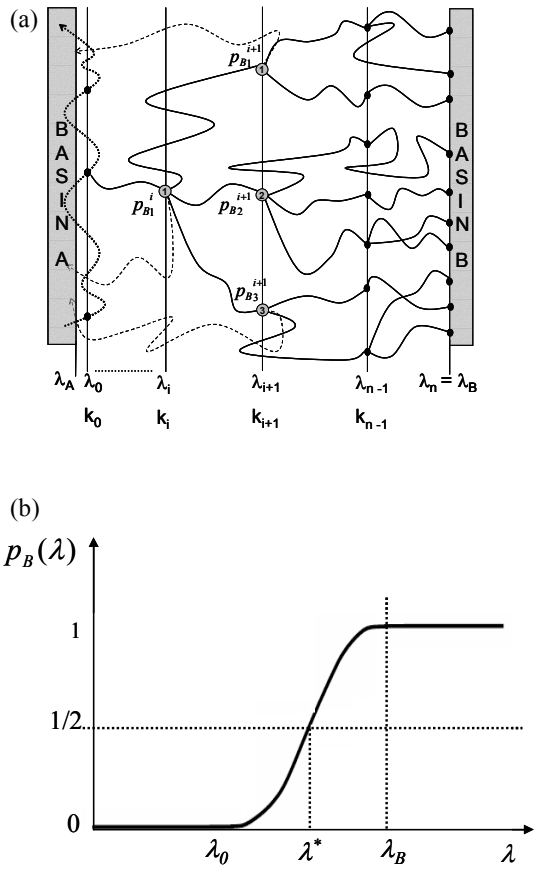

Figure 1.1 A schematic view of the generation of branched paths (thick lines) using the branched growth sampling method (BG). (a) The first stage involves the simulation run in the A basin shown by a dotted line. Starting points for the subsequent generation of branched paths are marked with a black circle at λ_0. The second stage corresponds to the trial runs (k_i) fired from λ_i ; those that reached the next λ_{i+1} interface are shown by a thick line and those which failed to reach λ_{i+1} are shown by a dotted line. In this example, k_i=4, k_{i+1}=3, and k_{i+2}=2. $p_{B_j}^{n} = 1$ for all points collected at $\lambda_{n=B}$. $p_{B_j}^{i+1}$ for the points j at λ_{i+1} are estimated from Eq. (5) as follows: $p_{B_1}^{i+1} = 1/3 * [2/2 + 1/2]$, $p_{B_2}^{i+1} = 1/3 * [1/2 + 1/2 + 2/2]$, and $p_{B_3}^{i+1} = 1/3 * [1/2 + 2/2]$. The $p_{B_1}^{i}$ value for the point 1 at λ_i is then obtained from: $p_{B_1}^{i} = 3/4 * \left[p_{B_1}^{i+1} + p_{B_2}^{i+1} + p_{B_3}^{i+1}\right]/3 = 1/4 * [1/2 + 2/3 + 1/2]$. (b) A sketch of the committor function, $p_B(\lambda)$, in one dimension. $p_B \approx 0$ for points inside and near basin A because trajectories started thereof end in A; $p_B \approx 1$ for points in or near basin B because trajectories started thereof end in B. The statistical modeling of $p_B(\lambda)$ is restricted to the intermediate, transitional region of λ; the dashed line illustrates a sample linear fit. The configurations for which $p_B(\lambda^*) \approx ½$ are called transitions states.

paths. The flux $\overline{\Phi}_{A,0} / \overline{h}_A$ in Eq. (1) is obtained from the simulation run in basin A, by dividing the total number of crossing configurations at λ_0 by the total length of this run.

The characteristic transition pathways are obtained beginning with the collection of trials which arrive at $\lambda_B=\lambda_n$ from λ_{n-1} and tracing back the sequence of connected partial paths which link them to region A. For a complete description of the theoretical background of the FFS-type algorithm and the path sampling schemes, we refer the readers to Ref. 2.

III. Methods

A. p_B history from FFS-type simulation

As the trajectories harvested by FFS-type path sampling evolve in phase-space, the stored points at each interface evolve in configuration space. Hence, each stored point has an intrinsic committor probability to reach the final region B. To extract p_B, the phase-space coordinates of the system must be stored for all points along all the trial runs which successfully reach λ_{i+1} from λ_i. One must also store information on the connectivity of the partial paths; i.e., by annotating each successful trial from λ_i to λ_{i+1} with an index that describes its initial point at λ_i. The BG scheme is ideal for tracking the p_B history, because from each stored point at λ_i one can estimate a p_B value from the outcome of the k_i branch generation trials started therein. Furthermore, the TPE is sampled with better resolution in the phase-space region close to TSE and B, where the transition paths are more branched.

The probability of each point at λ_i to reach the next interface λ_{i+1} before the initial region A, $p_j^i\left(\lambda_{i+1} \mid \lambda_i\right)$, can be easily obtained from:

$$p_j^i\left(\lambda_{i+1} \mid \lambda_i\right) = \frac{N_j^i}{k_i} \qquad (4)$$

where N_j^i is the number of points reaching λ_{i+1} from point j at λ_i. The values of $p_j^i(\lambda_{i+1} \mid \lambda_i)$ are then used to calculate the B-committor probability of point j at λ_i, $p_{B_j}^{\ i}$:

$$p_{B_j}^{\ i} = p_j^i(\lambda_{i+1} \mid \lambda_i) \frac{\sum_{m=1}^{N_j^i} p_{Bm}^{\ i+1}}{N_j^i} = \frac{\sum_{m=1}^{N_j^i} p_{Bm}^{\ i+1}}{k_i} \qquad (5)$$

Eq. (5) states that the $p_{B_j}^{\ i}$ of point j at λ_i can be expressed as the product of the probability $p_j^i(\lambda_{i+1} \mid \lambda_i)$ that a trajectory initiated from this point reaches λ_{i+1} and the average $p_{B_j}^{\ i+1}$ of all points at λ_{i+1} that connect with that state j at λ_i. Once the FFS-type simulation is complete, $p_{B_j}^{\ i}$ values are obtained by following the trials that reached B back to λ_{n-1}, then following their connected partial paths back to λ_{n-2}, and so on back to A. As illustrated in Figure 1.1(b), each point at λ_n has $p_{B_j}^{\ n} = 1$. For each point at λ_{n-1}, the committor probability to B is simply given by $p_{B_j}^{\ n-1} = p_j^{n-1}(\lambda_n \mid \lambda_{n-1})$. Then, $p_{B_j}^{\ n-1}$ values at λ_{n-1} are used to estimate $p_{B_j}^{\ n-2}$ at λ_{n-2} using Eq. (5), and so on back to A. The FFS-type scheme distributes points in the full range of p_B values, and any reaction coordinate obtained from the p_B history will apply at every p_B along the transition pathways.

B. Reaction coordinate from p_B history

In general, we are interested in constructing the best possible reaction coordinates from a few collective variables. Therefore, the goal is to identify the simplest order parameter that not only measures the progress of the reaction but is also useful to characterize the reaction dynamics. Simple reaction coordinates are better for the description of the system's dynamics, but the challenge is to determine which variables are important. Our strategy is to propose m candidate collective variables

from which the model is constructed. The model is first tested to determine the variables that are significant for the fitting of the p_B data. These variables are then used to construct a second simpler model. In other words, this algorithm finds the best reaction coordinate as a function of few candidate variables that are significantly correlated with p_B.

The approach presented here is related to the maximum likelihood "ML" method recently proposed by Peters and Trout[19] for obtaining reaction coordinates but we use a different method for sampling p_B (FFS rather than TPS) and for finding the model (LSE rather than ML, though LSE is a type of ML estimation). It is, however, beyond the scope of this work to compare the performance of both methods. The basis of both approaches is that p_B can serve as a guide to search efficiently the space of physically meaningful variables. Our method begins by harvesting an ensemble of typical trajectories for the transition between two stable states. This ensemble is obtained from a FFS-type simulation using an initial guess for the order parameter. As described in Sec. III A, the p_B value for each of the points crossing the interfaces is extracted from their path connectivity. Reaction coordinate candidates are then tested based on this p_B history by fitting the p_B data to a model, $\boldsymbol{q(x)}=q_1, q_2,\ldots,q_m$, that depends on m collectives variables, which are considered potentially useful descriptors. The collective variables are evaluated at each stored point where the p_B value is known.

A simple model of the reaction coordinate is favored in which the response variable p_B may be related to m collective variables:

$$\lambda(q) = p_B(q) = \sum_{k=1}^{m} \beta_k q_k + q^T A q + \beta_0 + \varepsilon \qquad (6)$$

This kind of model is called a *response surface model* with m regressor variables.[25] The parameters β_j, j=0,1,…,m, are called the regression coefficients and absorb the units from the collective variables, so the reaction coordinate is dimensionless. β_j represents the expected change in λ per unit in q_j when all the remaining independent variables q_i (i≠j) are held constant. The β_0 parameter allows the reaction coordinate to shift so the transition states are located at $\lambda(q)$=½. Interactions between collective variables are included by the cross quadratic term in Eq. (6) where **A** is a matrix of adjustable parameters.

The use of a model like that of Eq. (6), however, may not be sufficiently flexible to describe the shape of the $p_B(\lambda)$ function. As illustrated in Figure 1.1(b), even if $p_B(\lambda)$ is nearly a linear function over some intermediate range of λ, it will necessarily have a $p_B = 0$ plateau for λ in the A basin and a $p_B = 1$ plateau for λ in the *B* basin. Hence, it is convenient to perform the fitting of the model (6) to the p_B history data disregarding configurations with $p_B = 0$ and $p_B = 1$. In this way, the optimized order parameter obtained from our method will only be suitable for estimating p_B values for configurations that lie in the phase space between the two stable states. Because the model surface is not bound to lie in [0,1], all states with a $p_B \leq 0$ can then be enclosed together, defining the initial basin of attraction *A* with $p_B = 0$. Likewise, the region *B* can be defined by enclosing all states with $p_B \geq 1$ and assigning them a nominal $p_B = 1$. This procedure avoids problems that could result from the fitting of p_B history data containing a large number of states with $p_B = 0$ or 1; in such a case, the resulting p_B model would mainly try to fit the plateau regions of the two stable states.

The regression coefficients in Eq. (6) are determined (i.e., model fitting) by standard least-squares estimation (LSE). The method of LSE chooses the β*'s* so that the sum of the squares of the errors, ε (i.e., uncorrelated random variable error), is minimized.[25] The adequacy of the model fit is then checked by performing an analysis

of variance (ANOVA) that will determine if there is a statistically significant correlation between the response variable p_B and the subset of the collective variables, $\boldsymbol{q(x)}=q_1, q_2,\ldots,q_m$. The ANOVA involves first the partitioning of the total sum of squares SS_T into a sum of squares due to the model (or the regression) $SS_R(\beta_j)$ and a sum of squares due to residual (or error) SS_E, and then the testing of the null hypothesis H_0: $\beta_1=\beta_2=\ldots=\beta_m=0$ by computing:[25]

$$F_0 = \frac{SS_R(\beta_j)/k}{SS_E/(n-k-1)} = \frac{MS_R}{MS_E} \qquad (7)$$

here k denotes the degrees of freedom (*df*) for the model, *n-k-1* are the degrees of freedom for the residual, MS_R is the mean square for the model, MS_E is the mean square for the residuals, and n is the total number of observations for p_B history. H_0 is rejected if F_0 exceeds $F_{\alpha,k,n-k-1}$ or alternatively if the *P*-value for the statistics F_0 is less than α (here chosen to be to 0.05 for a 95% confidence interval). The *P*-value of an observed *F*-value is the probability that, given that the null hypothesis is true, the random variable being tested will assume an *F*-value equal or smaller than the observed F_0.[25] Individual regression coefficients are also tested to determine which collective variables are significant for the model description. Adding a variable to the regression model often causes the SS_R to increase and the SS_E to decrease. However, we must decide whether the increase in the SS_R is sufficient to warrant keeping the additional variable in the model. To this end, a null hypotheses H_0: $\beta_j=0$ for testing the significance of any individual regression coefficient is evaluated by a partial *F*-test:[25]

$$F_o = \frac{SS_R(\beta_j \mid \beta_{i\neq j})/1}{MS_E} \qquad (8)$$

where $SS_R(\beta_j \mid \beta_{i\neq j}) = SS_R(\beta) - SS_R(\beta_{j\neq i})$ is the regression sum square due to β_j given that $\beta_{i\neq j}$ coefficients are already in the model. H_0 is rejected if F_0 exceeds $F_0 > F_{\alpha, 1, n-k-1}$ or alternatively if the P-value for the statistics F_0 is less than α. The new order parameter (or reaction coordinate) describes a hyper-plane in the space of the significant collective variables $\{q_i\}$ for the p_B surface response. Moreover, because the model fitting uses information from the full range of p_B values, the reaction coordinate should apply to all path regions where p_B values were collected.

Our approach is different from typical approaches,[20,22,23] in which trial reaction coordinates are first proposed and then tested. In those approaches, a good reaction coordinate is found by first selecting combinations of order parameters (based on physical intuition) and then testing them for their ability to describe transition states via additional dynamical simulations. Unlike these approaches[20,22,23] but in analogy with Peters and Trout[19] approach, our method allows additional variables to be tested without sampling new trajectories or configurations on a constraint surface. Any such new variable is simply calculated at each of the stored points obtained from the initial FFS-type simulation. The major difference with Peters and Trout[19] method is that their approach uses maximum likelihood estimation (MLE) to seek the probability distribution function (PDF) that makes the observed p(TP|x) or p_B(x) data most likely. However, application of MLE requires knowledge of the PDF of the data under each proposed model.[26] To this end, they used the aimless shooting algorithm to generate most shooting points (i.e. p_B history) near the unknown p_B=½ surface.[19] This allowed them to assume a PDF model for the p(TP|x) or p_B(x) data which was then used to calculate the likelihood of the model given the shooting data. In our approach, since we do not know the PDF for the FFS-type simulation data, LSE is more convenient for the estimation of parameters in Eq. (6).[26]

The entire process described here for obtaining an optimized order parameter can be repeated, so that the current best model is used to generate additional p_B data to be LSE-fitted to model (6) and get then a better estimate of the reaction coordinate, and so on until some convergence criterion is met. For the simple systems analyzed in this work, however, one iteration of the process was deemed sufficient.

IV. Results

A. Example 1: Analytical potential energy surfaces

Here we apply the FFS-LSE method to find reaction coordinates and transition paths in two simple systems involving a particle moving on a two-dimensional potential energy surface. These model surfaces provide a convenient test bed to validate the proposed method because the stable states and transition states are visually identifiable and known a-priori.

The first model system is defined by an energy potential:

$$V_1(x,y) = \frac{[4\left(1-x^2-y^2\right)^2 + 2\left(x^2-2\right)^2 + \left((x+y)^2-1\right)^2 + \left((x-y)^2-1\right)^2 - 2]}{6} \qquad (9)$$

This model surface has been used previously to test transition path sampling methods.[10] A contour graph of this energy landscape is shown in Figure 1.2. The potential surface shows two well defined minimum at (-1, 0) and (1, 0).

The initial and final regions were defined by circles centered at the minima, with a radius of 0.2 and 0.3, respectively. The kinetics of the system was simulated using MC dynamics at $\beta = 1/k_B T = 8$ .The move set entailed the sampling in each dimension of a normal distribution centered on the current point:

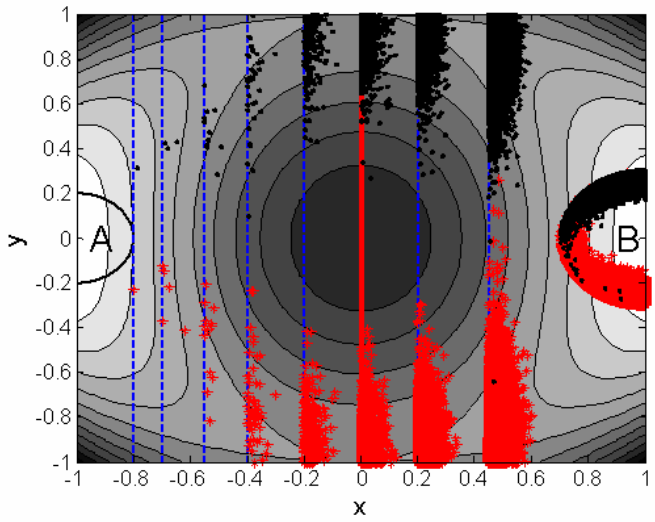

Figure 1.2 Contour graph of $V_1(x,y)$. The color scheme changes from highest (black) to lowest (white) elevations. The initial and final regions are shown by the circles labeled A and B, respectively. The initial guess λ vector for the FFS-type simulation is shown by dotted lines (blue). The thick line (red) shows the TS dividing surface. Two branched paths are also shown: (● black, and * red). These state points were used to determine the p_B history for the reaction coordinate estimation.

$$\eta(x_\alpha, x_{\alpha'}) = \frac{1}{\sigma\sqrt{2\pi}} e^{\left[-(x_\alpha - x_{\alpha'})^2 / 2\sigma^2\right]} \qquad (10)$$

where σ=0.04 is the standard deviation, chosen according to the distance the particle is expected to travel due to diffusion. The starting points for the FFS-type simulations were randomly sampled from inside the region *A*. The initial guess of order parameter for the BG simulations was given by:

$$\lambda_i = \begin{cases} x & i \neq n = B \\ 1 - 0.5\sqrt{(x-1)^2 + y^2} & i = n = B \end{cases} \qquad (11)$$

Accordingly, we defined $\lambda_0 = x = -0.80$ and the final region by taking $\lambda_B = \lambda_n = 1 - 0.5\sqrt{(x-1)^2 + y^2} = 0.85$. Note that we used the same definitions for the initial and final region as in previous works.[10] We used eight interfaces to partition the phase space (n=8) positioned at λ_i ($0 \le \lambda_i \le n$): $\boldsymbol{\lambda}(x)$ = {-0.80, -0.70, -0.55, -0.40, -0.20, 0.00, 0.20, 0.45, 0.85}. The number of trials per point at λ_i was k_i = {100, 40, 20, 10, 10, 5, 4, 4} ($0 \le i < n$). Figure 1.2 also shows the $\boldsymbol{\lambda}$ vector surface and typical transition paths obtained from two independent FFS-type simulations.

Since for this model system the transition state (TS) dividing surface is known a priority (dark/red thick line in Figure 1.2), we can directly test if our proposed approach leads to the estimation of an optimal order parameter (i.e., p_B surface) having a value of ½ on this TS dividing surface. The p_B history was obtained from the TPE by the method outlined in Sec. III A and fitted to a tentative regression model, including two collective variables (i.e., *x* and *y* coordinate) and an interaction term (*xy*):

$$\lambda(x, y) = \beta_1 x + \beta_2 y + \beta_3 xy + \beta_0 \qquad (12)$$

Table 1.1 LSE parameters and analysis of variance for the reaction coordinate model of the two-dimensional energy surface $V_1(x,y)$.

Source	Sum of Squares	df	Coefficient	Mean Square	F-value	P-value
Model	405.2	3		136.1	5669	< 0.0001
X [β_1]	405.1	1	0.856	405.1	16878	< 0.0001
Y [β_2]	0.0543	1	0.002	0.0543	2.26	0.1328
XY [β_3]	0.0325	1	0.010	0.0325	1.35	0.2453
Constant [β_0]			0.503			
Residual	518.5	21894		0.024		
Lack of fit	517.8	21870		0.024	0.83	0.7765
Pure Error	0.70	24		0.029		
Corr. Total	923.7	21897				

Since there is no curvature in the system (i.e., TS dividing surface is represented by a straight line), polynomial terms of higher degree (e.g., x^2 and y^2) were excluded from the model. Table 1.1 shows the LSE parameters and ANOVA for the reaction coordinate model of this system. The *P*-value in Table 1.1 for the *F* statistic of the model [Eq. (12)] is very small, indicating that at least one of the three variables has a nonzero regression coefficient. Table 1.1 also reports the *F* statistic for the test of lack of fit. This test partitions the SS_E into two components, $SS_E = SS_{PE} + SS_{LOF}$, where SS_{PE} is the sum of squares due to pure error and SS_{LOF} is the sum of squares due to lack of fit. SS_{PE} is obtained by computing the corrected sum of squares of the repeated observations at each level of the collective variables ($\boldsymbol{q}$) and then pooling over the *m* levels of $\boldsymbol{q}$.[25] Therefore, to test for lack of fit, we would compute the test statistic F_0 and conclude that the regression function is not valid if $F_0 > F_{\alpha,,m-k,n-m}$. Since the *P*-value for the *F* statistic of lack of fit in Table 1.1 is very high, we conclude that the model is significant. The upper portion of this table also gives the LSE of each parameter, the partial *F*-value statistic, and the corresponding *P*-value. As expected, the partial *F*-test shows that the *x* coordinate is the only important collective variable for the description of the model. Therefore, the reaction coordinate for this particular energy surface only depends on the *x* coordinate (i.e., *P*-value < 0.0001). A second LSE was performed by considering in the reaction coordinate model the *x* regressor only:

$$\lambda(x) = p_B(x) = 0.504 + 0.853x\,. \qquad (13)$$

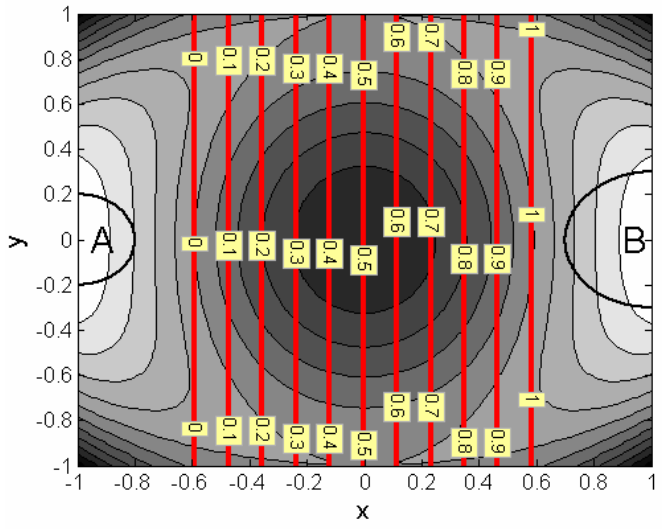

Figure 1.3 Estimated reaction coordinate iso-lines for model system $V_1(x,y)$. Thick lines (red) are the predicted p_B committors from LSE. The committor values appear as labels over the lines. The contour of the free energy surface $V_1(x,y)$ is shown only for reference. The color scheme changes from highest (black) to lowest (white) elevations.

Figure 1.3 shows the p_B contours for the optimal reaction coordinate given by Eq. (13). The optimal λ gives a reaction coordinate surface with a p_B ~½ when x=0; the energy contours in this figure are shown only to illustrate that the method identifies the correct TS dividing surface ($\lambda(x) = p_B(x) = ½$). Note that those states below the p_B =0 committor surface simply commit to the initial basin *A*. Likewise, all those states above the p_B =1.0 surface simply commit to the final region *B*. This example illustrates that our method is successful in identifying an optimal order parameter [i.e., $\lambda(\boldsymbol{q})$] and the correct transition state at $\lambda=p_B$ =½.

A second more complicated energy potential surface was also used to test our method:[27]

$$V_2(x,y) = -e^{-\left((x-1)^2+y^2\right)} - e^{-\left((x+1)^2+y^2\right)} + 5e^{-0.32\left(x^2+y^2+20(x+y)^2\right)} + \frac{32}{1875}\left(x^4+y^4\right) + \frac{2}{15}e^{-2-4y} \quad . \tag{14}$$

A contour plot of the potential is given in Figure 1.4. This potential surface shows the characteristics of a bistable potential with two minima at (-0.98, 0.06) and (0.96, 0.06) and one saddle point at (-1.96, 1.96). Note that if we were to assume that x is the only important variable and integrate out all the other degrees of freedom (in this case y), then the free energy maximum (along x) would not be located at the saddle point of the surface. Further, to go over the barrier a particle would first have to move in the opposite direction. Therefore, the optimal reaction coordinate that describes the system's dynamics is expected to depend of both coordinates (x and y) and be a nearly lineal function of them in order to describe correctly the TS dividing surface (dark/red thick line in Figure 1.4). Hence, the initial guess of order parameter for the FFS-type simulation was assumed to be:

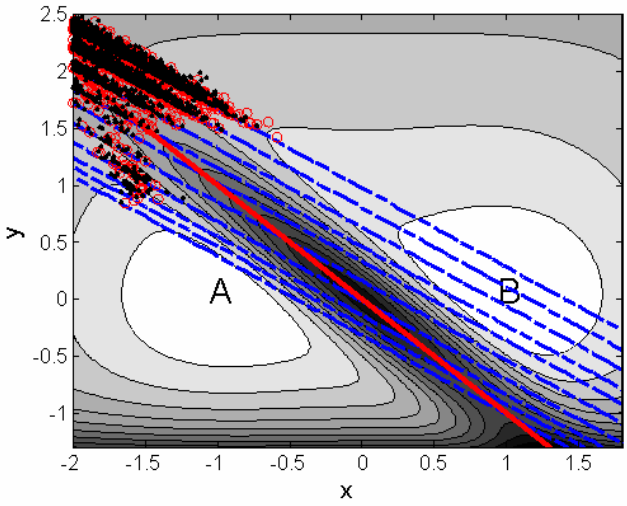

Figure 1.4 Contour graph of $V_2(x,y)$. The color scheme changes from highest (black) to lowest (white) elevations. The initial and final regions are labeled as A and B, respectively. The initial guess λ vector for the FFS-type simulation is shown by dotted lines (blue). The thick line (red) shows the TS dividing surface. Two branched paths are also shown: (● black and o red). These state points were used to determine the p_B history for the reaction coordinate estimation.

$$\lambda(x, y) = 0.268 + 0.807x + 1.164y \,. \qquad (15)$$

This $\boldsymbol{\lambda}$ surface was chosen such that it has a p_B value different from ½ at (0, 0) and the saddle point. We expected that in this case our approach will produce an improved reaction coordinate with the correct regressor coefficients, leading to a slope change of the initial λ iso-contours and a displacement to $\lambda(\boldsymbol{q})$=½ at the TS dividing surface. Figure 1.5 shows the $\boldsymbol{\lambda}$ vector surface and typical transition paths obtained from the BG simulations. We used nine interfaces to partition the phase space (n=9) positioned at λ_i ($0 \le i \le n$): $\boldsymbol{\lambda}(x,y)$ ={-0.10, 0.00, 0.10, 0.25, 0.45, 0.65, 0.80, 1.00, 1.20, 1.40}. The number of trials per point at λ_i was k_i = {100, 40, 20, 10, 10, 5, 4, 4} ($0 \le i < n$). The kinetics of the system was also studied using MC dynamics at $\beta = 1/k_B T = 6$ and the same move set defined by Eq. (10). The p_B history was obtained from the TPE and fitted to the tentative regression model of Eq. (12). Table 1.2 shows the LSE parameters and ANOVA for this case. Since the P-value for the test of lack of fit is high, the regression model can be used to describe the variability of the data with a least one of the three collective variables having a nonzero regression coefficient (i.e., P-value< 0.0001 for F-value of the model). As expected, the optimized reaction coordinate for this particular energy surface depends on both coordinates:

$$\lambda(x, y) = p_B(x, y) = 0.616 + 1.153x + 0.941y - 0.109xy \,. \qquad (16)$$

Figure 1.5 shows the p_B contours for this optimal reaction coordinate. The energy contours in this figure are shown only to illustrate that the method identifies the correct TS dividing surface ($\lambda(x, y) = p_B(x, y) = ½$).

B. Example 2: Genetic switch

This case illustrates the use of our approach for calculating reaction coordinates for a non-equilibrium, rare event process. The system is a set of chemical

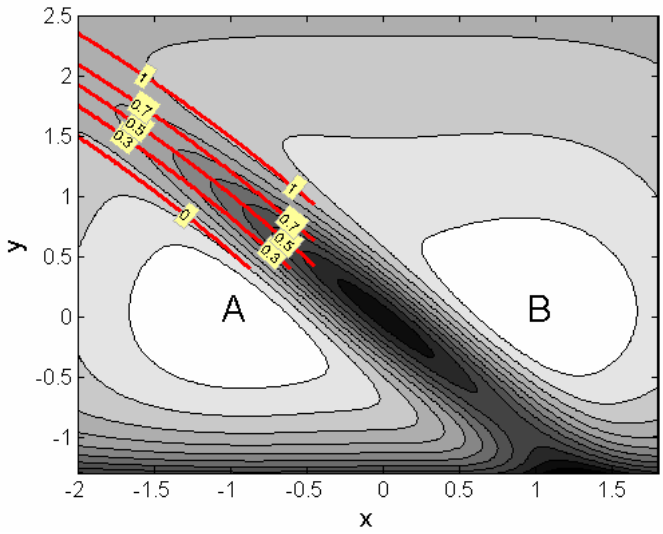

Figure 1.5 Estimated reaction coordinate iso-lines for model system $V_2(x,y)$. Thick lines (red) are the predicted p_B committors from LSE, whose values appear as labels. Contour of the free energy surface $V_2(x,y)$ are shown only for reference. The color scheme changes from highest (black) to lowest (white) elevations.

Table 1.2 LSE parameters and analysis of variance for the reaction coordinate model of the two-dimensional energy surface $V_2(x,y)$.

Source	Sum of Squares	df	Coefficient	Mean Square	F-value	P-value
Model	6558	3		2186	99361	< 0.0001
X [β_1]	96.49	1	1.153	96.49	4386	< 0.0001
Y [β_2]	6364	1	0.941	6364	289268	< 0.0001
XY [β_3]	97.45	1	-0.109	97.45	4430	< 0.0001
Constant [β_0]			0.616			
Residual	4700	210530		0.022		
Lack of fit	4695	20825		0.023	1.00	0.5123
Pure Error	5.33	228		0.023		
Corr. Total	11258	210533				

reactions for a model symmetric bistable genetic switch whose dynamics do not obey detailed balance. This system has been used previously to test FFS sampling methods.[2,18] The switch consists of a piece of DNA containing two genes A and B, and an operator site (O). The two genes encode for proteins A and B, which can dimerize and bind to the operator site in their homodimer form. However, only one dimer can be bound to DNA (i.e., operator site) at any time, controlling protein production. That is, when A_2 is bound to O, A is produced at rate k, but B is not produced. Likewise, when B_2 is bound to O, B is produced at rate k, but A is not produced.

The FFS-type simulations were carried out using the Gillespie algorithm,[5] which is a widely used Kinetic Monte Carlo scheme for propagating chemical reactions. A description of the reactions involved and rates constants for this genetic switch are given in Table 1.3.[2,18]

In previous studies, the order parameter was chosen to be $\lambda = N_B - N_A$, where N_A is the total number of A proteins, and N_B the total number of B proteins.[2,18] The probability distribution $P(\lambda)$, shown in Figure 3(b) of Ref. 18, shows that there are two well defined steady states. Therefore, we anticipate that this order parameter should describe well the reaction coordinate, leading to an efficient sampling of pathways. To show that our approach leads to the estimation of an optimal order parameter, we chose a different definition of λ as initial guess for the BG simulations, namely: $\lambda = -N_A$. We defined the phase space region A by taking $\lambda \leq$ -24, and B by $\lambda \geq$ -4. The phase space was partitioned with nine interfaces (n=9) positioned at λ_i ($0 \leq \lambda_i \leq n$): $\boldsymbol{\lambda}(N_A)$ ={-24, -22, -20, -18, -15, -12, -10, -8, -6, -4}. The number of trials per point at λ_i was k_i = {100, 20, 20, 10, 10, 8, 6, 6, 4} ($0 \leq i < n$). The p_B history was obtained from the TPE and fitted to the tentative regression model with two regressor coefficients for N_A and N_B and an interaction term [see Eq. (12)]. Table 1.4 summarizes the LSE parameters and ANOVA for the reaction coordinate model of

Table 1.3 Reactions involved for the genetic switch. Forward and backward rate constants k_f and k_b are also given. [2,18]

Protein A	Protein B	k_f	k_b
$2A \leftrightarrow A_2$	$2B \leftrightarrow B_2$	$5k$	$5k$
$O + A_2 \leftrightarrow OA_2$	$O + B_2 \leftrightarrow OB_2$	$5k$	k
$O \rightarrow O + A$	$O \rightarrow O + B$	k	
$OA_2 \rightarrow OA_2 + A$	$OB_2 \rightarrow OB_2 + B$	k	
$A \rightarrow \varnothing$	$B \rightarrow \varnothing$	$0.25k$	

Table 1.4 LSE parameters and analysis of variance for the reaction coordinate model of the genetic switch.

Source	Sum of Squares	df	Coefficient	Mean Square	F-value	P-value
Model	2890	3		963.4	34405	< 0.0001
N_A [β_1]	1374	1	-0.038	1374	49061	< 0.0001
N_B [β_2]	1516	1	0.039	1516	54154	< 0.0001
$N_A N_B$ [β_3]	0.0857	1	0.0002	0.0857	3.06	0.0803
Constant [β_0]			0.500			
Residual	605.9	21664		0.0280		
Lack of fit	38.81	1340		0.0286	1.02	0.3052
Pure Error	567.1	20324		0.0280		
Corr. Total	3496	21667				

this system. The ANOVA shows that the regression model can be used to describe the variability of the data with statistically significant regression coefficients for the collective variables N_A and N_B (i.e, P-value < 0.0001 for partial F-value statistic). Since the P-value for the test of lack of fit is high, the model is deemed significant. Figure 1.6 shows the $\boldsymbol{\lambda}$ response surface for the optimal order parameter projected into N_A and N_B coordinates:

$$\lambda(N_A, N_B) = p_B(N_A, N_B) = 0.500 - 0.038N_A + 0.039N_B. \quad (17)$$

The free energy landscape in this figure is shown only to illustrate that the method identifies the TS dividing surface ($\lambda(N_A, N_B) = p_B(N_A, N_B) = ½$), that projects onto a line with ~45° slope and origin at (0, 0). Again, as expected, our approach generates an optimized reaction coordinate with similar and oppositely signed regressor coefficients for N_A and N_B. The constant β_0 shifts $\lambda(\boldsymbol{q})$=½ at the TS dividing surface. This example illustrates the applicability of the FFS-LSE approach to an intrinsically non-equilibrium stochastic process (for which some alternative approaches are unsuitable).

C. Example 3: Lattice protein folding

Recently, we applied a FFS-type approach[3] for calculating transition rate constants and for sampling folding paths of a simple cubic lattice protein. The FFS algorithm generated trajectories for the transition between the unfolded and folded states, which were then used to obtain the TSE and the properties that characterize these intermediates. Furthermore, we showed that for this simple system, the fraction of native contacts (Q) is a good order parameter that describes the dynamical bottleneck between the folded and unfolded stable states. Now we explore the possibility that our FFS-LSE method could lead to an even better order parameter for

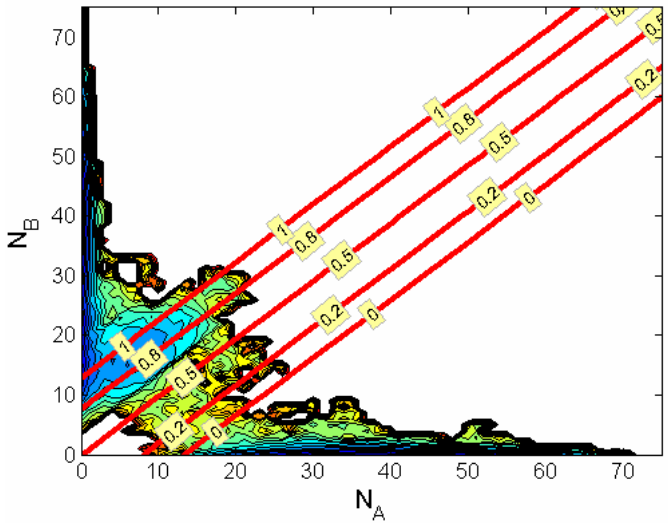

Figure 1.6 Estimated reaction coordinate iso-lines for the genetic switch model. Thick lines (red) are the predicted p_B committors from LSE, whose values appear as labels. Contour of the free energy landscape (N_B vs. N_A) are shown only for visual reference. The color scheme changes from highest (gray/red) to lowest (black/blue) elevations. The TS dividing surface ($\lambda(N_A, N_B) = p_B\ (N_A, N_B) = ½$) passes through a line at a ~45°angle, with origin at $N_B = N_A$=0.

this system, allowing a higher computational efficiency for FFS-type simulations and the identification of the TS dividing surface and the TSE without the need to perform additional committor simulations.

The 48-mer protein model adopted here has a unique native structure, that is, a compact structure whose energy is minimum $-E_{min}= 20.24\ k_BT$ and has $NNC = 57$ specific or "native" segment-segment contacts. In addition, it exhibits a fast and stable protein-like folding with a two-state (unfolded-folded) transition.[28] Further details on the model protein, including its structure, thermodynamics and kinetic behavior are given elsewhere.[3,28,29]

The folding kinetics of the system was simulated in free space at the corresponding *bulk transition temperature* $T_f = 0.27$.[3] Figure 1.7 shows the contour plot of the free-energy landscape for this system projected over the plane of energy (E) and "nativeness", as given by the number of native contacts (NC); the unfolded state (λ_A) was defined by a circle centered at (NC=0, E=0) and radius of 9. The starting points for the FFS simulations were randomly sampled from inside this region A. The folded state was defined by taking $\lambda_B=\lambda_n= NNC$. The nativeness was used as initial guess of the order parameter (i.e., $\lambda=NC$) to partition the phase space between the unfolded and folded states for the BG simulations:

$$\lambda_0 = \sqrt{NC^2 + E^2} = 9 \qquad (18)$$

$$\lambda_i = NC \qquad i > 0 \qquad (19)$$

We used twelve interfaces to partition the phase space (n=12), with interfaces positioned at λ_i ($1 \le i \le$ n): $\lambda(x)$ = {10, 12, 15, 20, 26, 29, 33, 37, 41, 46, 51, 57}.The number of trials per point at λ_i was $k_i = 10$ for $0 \le i < n$. The p_B history was obtained from the TPE by the method outlined in Sec. III A and fitted to a tentative regression

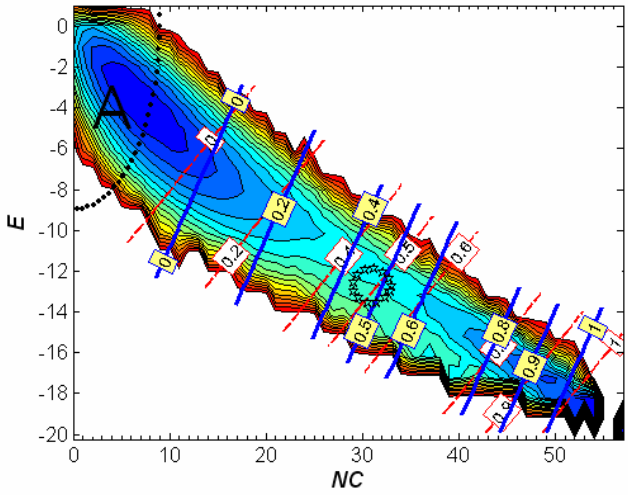

Figure 1.7 Estimated reaction coordinate iso-lines for the folding of the 48-mer lattice protein model at T_f=0.27. Thin dotted lines (red) are the predicted p_B committors from FFS-LSE with λ=NC as initial guess for the order parameter. Thick lines (blue) are the predicted p_B committors from FFS-LSE with λ=-E as initial guess for the order parameter. The committor values are shown as labels. Contour plot of the free energy landscape [energy (E) vs. nativeness (NC)] is shown only for visual reference. The color scheme changes from highest (red) to lowest (blue) elevations. The initial region is enclosed by the circle labeled A. The most probable visited transition state region is enclosed by the ellipse with center at (μ_{NC}=31, μ_E =-12.73).

model, including three collective variables: *NC*, *E*, and the square radius of gyration Rg^2, and interaction terms between these variables. The ANOVA for this model indicated that the linear terms for *NC* and *E* are the only significant ones. The insignificance of the Rg^2 term in the model is expected, because this variable does not help to make a clear distinction between the stable sates (e.g., structures from the unfolded basins can also be compact). Thus, a second LSE was performed only considering in the reaction coordinate model the *NC* and *E* regressors, see Table V:

$$\lambda(NC, E) = -0.404 + 0.017(NC) - 0.029(E). \qquad (20)$$

Table 1.5 reports a small and large *P*-value for the *F* statistic of the model and the test of lack of fit, respectively. Hence, the reaction coordinate model of Eq. (20) describes the variability of the of the p_B data with statistical significance. To show that our approach leads to the estimation of a good order parameter regardless of the choice of initial guess for the order parameter, we perform a separate BG simulation with λ=-*E* as the initial order parameter. To make this comparison consistent, we adopted identical definitions for regions *A* and *B*; namely, region *A* was defined by Eq. (18), and region *B* by taking $\lambda_B=\lambda_n= -E_{min}= 20.24\ k_BT$ (i.e., minimum energy for the folded state *NNC*=57). In this case, the phase space between the unfolded and folded states was partitioned with *n*=11 interfaces with λ_0 as in Eq. (18) and

$$\lambda_i = -E \qquad i > 0 \qquad (21)$$

positioned at λ_i ($1 \le i \le n$): λ={9,10,11,12,13,14,15,16,17,18,20.24}. The number of trials per point at λ_i was k_i = 10 for $0 \le i < n$. Again, the ANOVA in Table 1.6 indicates that the collective variables for *NC* and *E* are the most significant terms in the model:

Table 1.5 LSE parameters and analysis of variance for the reaction coordinate model of the lattice protein folding with $\lambda=NC$ as initial guess of the order parameter for the FFS-LSE method.

Source	Sum of Squares	df	Coefficient	Mean Square	F-value	P-value
Model	1582	2		790.8	82371	< 0.0001
***NC* [β_1]**	1565	1	0.017	1565	162566	< 0.0001
***E* [β_2]**	16.81	1	-0.029	16.81	1751	< 0.0001
Constant [β_0]			-0.404			
Residual	970.2	100752		0.0096		
Lack of fit	5.43	626		0.0086	0.90	0.9650
Pure Error	964.7	100126		0.0096		
Corr. Total	2552	100754				

Table 1.6 LSE parameters and analysis of variance for the reaction coordinate model of the **l**attice protein folding with $\lambda=-E$ as initial guess of the order parameter for the FFS-LSE method.

Source	Sum of Squares	df	Coefficient	Mean Square	F-value	P-value
Model	3185	2		1592	122026	< 0.0001
***NC* [β_1]**	3175	1	0.021	3175	244223	< 0.0001
***E* [β_2]**	9.87	1	-0.020	9.87	759.2	< 0.0001
Constant [β_0]			-0.427			
Residual	12965	99322		0.013		
Lack of fit	18.14	4096		0.004	0.31	1.000
Pure Error	1278	95226		0.013		
Corr. Total	4481	99324				

$$\lambda(NC, E) = -0.427 + 0.021(NC) - 0.020(E). \qquad (22)$$

The coefficients for the reaction coordinate model [i.e., Eq. (21) and (22)] obtained from the two independent BG simulations, each based on a different initial guess of the order parameter, are not identical but similar. Furthermore, both simulations consistently disregard the significance of the $NC \times E$ interaction and Rg^2 terms in the model. Figure 1.7 shows the λ response surface for the optimal order parameter from Eq. (20) and (22) projected onto the NC and E free-energy landscape. In both cases, the reaction coordinate model identifies the TS dividing surface [$\lambda(NC, E) = p_B(NC,E) = ½$], passing through $\mu_{NC} \approx 31$ (i.e., $Q \approx 0.54$) and $\mu_E \approx -12.73 k_B T$ with a positive linear slope. This value matches well the E value observed at the top of the energy barrier in Figure 6 of Ref. 3. The positive slope in the p_B surface indicates that states with lower E for the same NC have a higher probability to commit to the folded state, which is related to the formation of the correct folding foci.[3] Overall, we can state that the FFS-LSE approach leads to good approximations of a reaction coordinate no matter which order parameter is used for the initial phase space partition in the BG simulations. As indicated in Sec. V-B, more iterations of the process could be implemented where in each cycle the current optimized order parameter is used to obtain a new estimate for the reaction coordinate until satisfactory convergence; in this case, just one iteration was enough to get suitable results.

The optimized order parameter expressed as the p_B iso-committor surface leads to new definitions for the A and B regions. Because the model surface is not bound to lie in [0,1], all states with a $p_B \leq 0$ can be enclosed together, defining the initial basin of attraction A. Likewise, the region B is defined by enclosing all states with $p_B \geq 1$. It can be seen in Figure 1.7 that the new definitions of the stable states now more completely enclose the respective basins; these new definitions can then be seen as a refinement with respect to the original ones and a useful by-product of the method. To

show that this new definitions for the stable states do not affect the estimation of the average transition rate constant value ($k_{A\to B}$), we compared the $k_{A\to B}$ values for two choices of order parameters: (i) the optimized $\lambda = p_B(NC,E)$ as given by Eq. (20), and (ii) $\lambda = NC$. In both cases, the DFFS method (see Sec. II A and a brief description in the Appendix) rather than the BG algorithm was used, since the former is most robust to changes in the performance parameters (i.e., in the number of interfaces and of fired trial runs at each λ_i.), making it preferable for calculating transition rate constants.[1] For choice (i), the initial region A was defined by $\lambda_0 = p_B(NC,E) \le 0$ and the folded state region B was defined by $\lambda_B = \lambda_n = p_B(NC,E) \ge 1$. The optimized $\lambda(NC, E)$ of Eq. (20) is only suitable to estimate p_B values for configurations (x) constrained in the phase space between the two stable states, i.e.,:

$$p_B(x) = \begin{cases} \lambda(NC,E) = -0.404 + 0.017(NC) - 0.029(E) & 0 < \lambda(NC,E) < 1.0 \\ 0 & \lambda(NC,E) \le 0 \\ 1.0 & \lambda(NC,E) \ge 1.0 \end{cases} \qquad (23)$$

Nine interfaces were used to partition the phase space (n=9), with interfaces positioned at λ_i ($0 \le i \le$ n): $\lambda(NC,E)$ = {0.0, 0.1,0.2,0.3,0.4,0.5,0.6,0.8,1.0}.The number of trials at each λ_i was M_i=1000 for $0 \le i < n$. The statistics were accumulated during 100 blocks where a block consists of a complete DFFS calculation with N_0 starting points. These simulations gave $k_{A\to B}$ =8.33 ± 0.18 x 10^{-8}. For choice (ii) when $\lambda = NC$, we used twelve interfaces spaced in λ_i ($1 \le i \le$ n): $\lambda(NC)$ = {10, 12, 15, 20, 26, 29, 33, 37, 41, 46, 51, 57},$\lambda_0 = \lambda_A$ was defined as in Eq. (18) and $\lambda_0 = \lambda_B = NNC$. In this case the DFFS simulations gave $k_{A\to B}$ =8.74 ± 0.93 x 10^{-8}. As expected, the re-definition of the stable state regions implied by the reaction coordinate found by FFS-LSE does not affect the average transition rate constant. Note also that the statistical

error in the $k_{A\to B}$ estimate is smaller when the optimized order parameter is used to partition the phase space (which correlates with the increase in sampling efficiency to be discussed shortly). Furthermore, we envision that FFS-LSE could be a valuable tool for studying the dynamic in complex systems where the identification of the basins of attraction is more difficult to attain.

We compared our results for the NC distribution within the TSE with those for the protein folding in free space reported in Figure 9 of Ref. 3. To this end, histograms for *NC* and *E* were accumulated from the collection of configurations at the p_B contour of ½ ± σ [i.e., Eq. (20)] during the DFFS simulation (note that configurations from the original BG simulation would be too sparse for this comparison, since the points saved were not at $p_B \sim ½$). The statistics were accumulated during 50 blocks, each one starting from a different random point in region *A*. The standard error in the computed reaction coordinate model is $\sigma \approx 0.1$ (i.e., $MSE=\sigma^2$), see Table V, and so the $p_B = ½$ contour includes values in the range [0.4, 0.6]. In Figure 1.8, the *NC* and *E* distributions found from the TSE reaction coordinate iso-surface show Gaussian behavior with mean and standard deviation (μ ,σ) of (31, 2) and (-12.73, 0.83), respectively. These mean values of the intrinsic iso-committor surface identify the most populated intermediate regime of rapidly inter-converting conformations that are transient and do not accumulate (i.e., sparsely populated). Figure 1.7 also shows this intermediate region, represented by the ellipse with center at (μ_{NC}=31, μ_E =-12.73) and lengths of major and minor semiaxes σ_{NC}=2 and σ_E=0.83, respectively. These mean values match well the average position of the transition state (i.e., $\langle NC\rangle_{\mathrm{TSE}} = 31 \pm 3$) for the *NC* distribution determined from a committor analysis;[11] such a $\langle NC \rangle_{\mathrm{TSE}}$ corresponds to an average conformational energy $\langle E \rangle_{\mathrm{TSE}} \approx -12.68 \pm 1.02\ k_BT$.

Figure 1.8 also shows the normal distributions for *NC* and *E* obtained from the standard committor analysis.[11] In this analysis, a configuration is considered as a

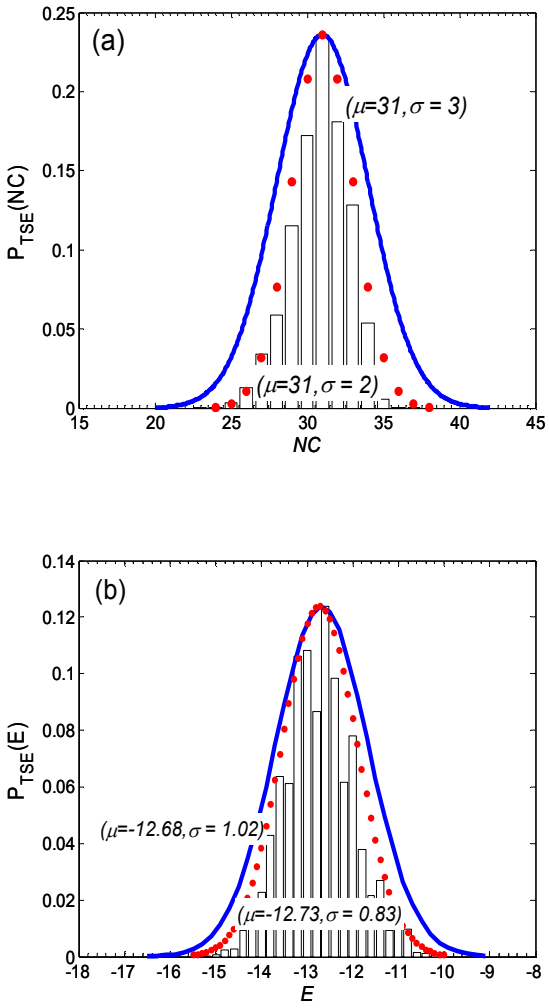

Figure 1.8 (a) Nativeness (NC) and (b) conformational energy (E) histograms for $p_B(NC,E) = -0.404 + 0.017(NC) - 0.029(E) = 1/2 \pm 0.1$ surface corresponding to the 48-mer lattice protein model at T_f =0.27. The histograms were accumulated from the states collected in the [0.4, 0.6] surface region during FFS-type simulation (DFFS scheme). The superimposed normal distribution (dotted/red line) that would result from sampling on the exact (μ,σ) surface is also shown. The exact Gaussian distribution obtained from committor analysis is shown by the solid line (blue).

member of the TSE if the interval of confidence $\left[p_B^{(N)} - \alpha\sigma^{(N)}, p_B^{(N)} + \alpha\sigma^{(N)}\right]$ includes the value ½. Here, N is the number of fleeting trajectories used to obtain $p_B^{(N)}$, and $\sigma^{(N)}$ is the standard deviation in the p_B estimate. In these simulations, the confidence level was fixed at α=1; more details of the committor analysis are given in Ref. 3. Note that the mean values μ for the *NC* and *E* distributions, approximated by the reaction coordinate iso-surface, are roughly the same as those of the Gaussian distributions obtained from the committor analysis.[11] Hence, the new optimized order parameter obtained from FFS-LSE can be considered as an appropriate reaction coordinate that describes properly the dynamical bottleneck between the two stable states for this lattice protein system.

Moreover, for a good reaction coordinate, the histogram for the approximate TSE should be closely centered around the characteristic committor value p_B=½.[11,30] Hence, the quality of the new order parameter from FFS-LSE [i.e., Eq. (20)] can be tested by collecting first states belonging o the TSE from a committor analysis,[11] and then calculating the p_B value for those states using the reaction coordinate model. Figure 1.9 shows the p_B histogram calculated from the model in Eq. (20) for 3674 states belonging to the TSE from a total of 100 branched paths analyzed. The adjusted normal distribution corresponds to μ_{pB} = 0.49 and σ_{pB} =0.17. Figure 1.9 also shows the exact normal distribution (μ=½, $\alpha\sigma_{pB}$ =0.17) that would result from sampling on the strict p_B=½ surface. Note that the committor distribution of the constrained TSE is peaked at p_B=½, providing a qualitative indicator that the optimized order parameter of Eq. (20) is a good reaction coordinator to describe the system's dynamics. Similar results are obtained if Eq. (22) were used instead of Eq. (20) in the analysis.

Although the expected value of the transition rate constant does not depend on the definition of λ, a good choice of order parameter is expected to improve the sampling efficiency.[1,2,17] Hence, the choice of λ will determine the optimum (i.e.,

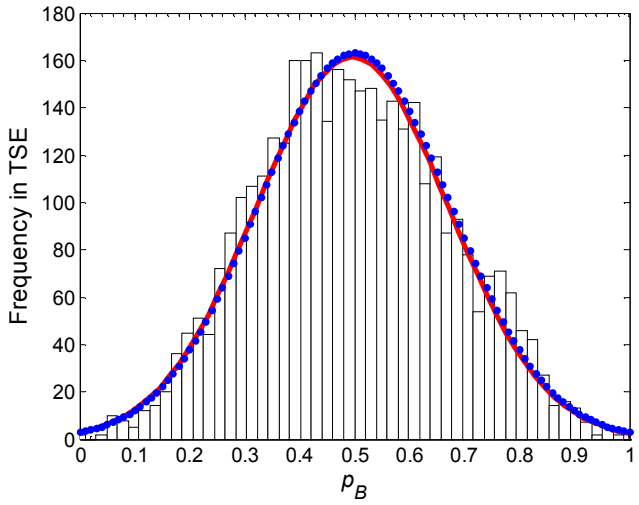

Figure 1.9 p_B histogram for the states collected at the TSE for the 48-mer lattice protein folding at T_f =0.27 from Ref. 3. p_B values were estimated from the reaction coordinate model: $p_B(NC,E) = -0.404 + 0.017(NC) - 0.029(E)$. The bin width is 0.01. The thick line (red) shows the normal distribution (μ_{pB} = 0.49 and σ_{pB} =0.17) obtained from the p_B histogram. The exact normal distribution (dotted/blue line) (μ=½, $\alpha\sigma$ =0.17) that would result from sampling on the exact p_B=½ surface is also shown. The standard error in the computed reaction coordinate model [Eq. (20)] is σ =0.1.

minimum) number of the interfaces (n) and fired trial runs (k) at each λ_i. Following Allen *et al.*, [1] the computational efficiency (ε) for FFS type simulations was defined as $1/[C\nu]$ where C represents the computational cost, estimated to be the average number of simulation steps (i.e., MC steps) per block, divided by the number of starting points at λ_0. The statistical error ν is defined to be the variance $V[P(\lambda_{n=B} \mid \lambda_0)]$ in the estimate of $P(\lambda_{n=B} \mid \lambda_0)$, per initial point at λ_0, divided by the square of the expectation value $E[P(\lambda_{n=B} \mid \lambda_0)]$:

$$\nu \equiv N_0 \frac{V[P(\lambda_n \mid \lambda_0)]}{P(\lambda_n \mid \lambda_0)^2} \qquad (24)$$

where N_0 is the number of starting points at λ_0 and $V[P(\lambda_n \mid \lambda_0)] = \overline{(P(\lambda_n \mid \lambda_0))^2} - \left(\overline{P(\lambda_n \mid \lambda_0)}\right)^2$. This definition of ε allow us to analyze the efficiency of the order parameter used in the FFS-type simulations in a systematic way.

As indicate before, DFFS is more robust than BG sampling to changes in the k and n parameters, remaining efficient even as k and n become large.[1] Therefore, we compared the ε of the DFFS scheme for the choice of two different order parameters: (i) $\lambda = p_B(NC,E)$ as given by Eq. (20), and (ii) $\lambda = NC$. Figure 1.10 shows a comparison of the computational efficiency obtained from DFFS simulations for the protein folding at T_f as a function of k. In these calculations, the same value of k was used for all interfaces: $M_i=k$ for all i. In both cases, the number of interfaces was fixed to n=9 because the performance of this algorithm appears to be insensitive to the choice of n as long as it is not too small.[1] To obtain the data in Figure 1.10, the nine interfaces

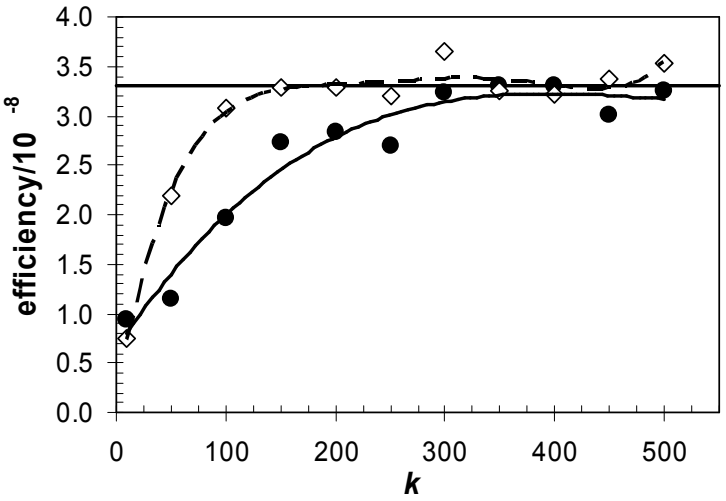

Figure 1.10 Measured efficiency ε for the 64-mer lattice protein folding at T_f=0.27: (--◊) $\lambda = p_B(NC,E)$ and (—●) $\lambda = NC$. Simulation results were obtained with 100 DFFS sampling blocks with N_0=1000 starting points per block. Interfaces were exponentially spaced for n=9. The efficiency plateau is marked by the solid horizontal line.

were exponentially spaced in λ_i ($1 \le i \le n$): $\lambda(NC,E) = p_B(NC,E)$ {-0.05,0.0,0.1,0.2,0.35,0.45,0.6,0.85, 1.0} and $\lambda(NC)$ ={10,12,15,20,26,29,37,46,57}. Note that each $\lambda_i(NC,E)$ corresponds to a p_B surface value such that it is crossed midpoint by $\lambda_i(NC)$ [in the NC vs. E plane]. The value of $\lambda_i(NC,E)$= -0.05 was adopted to ensure that the interfaces were similarly spaced in both cases (even though so this value is not meaningful in terms of p_B). The same definitions for the basins of attraction A and B were used to facilitate the comparison: region A is defined by Eq. (18) and the folded state region B by $\lambda_B = NNC$=57. For each ε estimate, simulations were carried out in a series of 100 blocks with N_0 starting points per block. Each block produces a result $P(\lambda_{n=B} \mid \lambda_0)$ for the probability to reach B, which is then used to find the variance between blocks $V\left[P(\lambda_n \mid \lambda_0)\right] = \overline{\left(P(\lambda_n \mid \lambda_0)\right)^2} - \left(\overline{P(\lambda_n \mid \lambda_0)}\right)^2$ where the overbar denotes block averaging.

We observed that as long as k is not too small, its value has little effect on ε. However, when the optimized order parameter is used to partition the phase space, an efficiency plateau (~3.3×10^{-8}) is reached for smaller k values, i.e. for $k \ge 150$ when $\lambda = p_B(NC,E)$ compared to $k \ge 300$ when $\lambda = NC$. Nonetheless, these results must be interpreted with caution, since several important factors are not included in the analysis leading to Figure 1.10. Firstly, our analysis does not consider the effect of the number and placement of the interfaces (though very closely spaced interfaces are likely to be computationally inefficient).[1] Secondly, our analysis neglects the effect of correlations between interfaces, though this is justified because neither of the FFS simulations shows a maximum in efficiency as a function of k (i.e., there is memory loss) for the choice of n=9 interfaces.[1] In short, we found that the optimized order parameter from FSS-LSE approach is robust to the choice of k, reaching a computational efficiency plateau at smaller k values than the typical choice of a good order parameter for this system. We envision that this increase in computational

efficiency for the FFS-type simulations will be more marked in more complex systems.

Overall this example illustrates that, in contrast to some previous methods,[9,11,21,22,24] FFS-LSE converges to an optimized order parameter that approaches the true reaction coordinate regardless the initial trial order parameter. Note also that our approach could avoid problems associated with typical approaches, in cases when a proposed reaction coordinate produced a too broad p_B histogram centered at the correct p_B value, indicating that additional candidate variables should be tested via further committor simulations. In our method, the data from the initial FFS-LSE simulation can be reused to screen the additional candidates, which only need to be evaluated at the saved points of the trajectories.

D. Mechanism details from FFS-type algorithm

Since the resulting reaction coordinate model from FFS-LSE corresponds to the p_B surface response, the TSE and hence the mechanistic details of the process can be readily obtained by only analyzing characteristics of the collective variables at the p_B contour of ½ $\pm \sigma$ (where σ is the desired level of statistical accuracy). To illustrate this, we determined the folding mechanism of a different lattice protein, a model 64-mer protein with the following sequence: KEKSTAGRVASGVLDSVACGVLGDID TLQGSPIAKLKTFYGNKFNDVEASQAHMIRWPNYTLPE. It also exhibits a single-domain native state with a maximum number of native contacts *NNC*=81 and E_{min}= -30.13k_BT, following an all-or-none transition between two clearly distinguishable states: the native and unfolded states. Folded structures and a detailed thermodynamic characterization for this model protein can be found elsewhere.[31-34] The center of the native structure of this protein presents a large and strong

hydrophobic core, which has made this model protein an ideal benchmark for kinetic studies of folding/aggregation dynamics.[31,32]

The FFS-LSE approach was applied to estimate the p_B surface for the folding kinetics in free space at the corresponding *transition temperature* $T_f = 0.27$ (data not shown). Figure 1.11 shows the contour plot of the free-energy landscape [E vs. NC] for this system. NC was used as initial guess of the order parameter (i.e., λ=NC) to partition the phase space between the unfolded and folded states in ten interfaces (n=10) positioned at λ_i ($0 \le i \le$ n): $\lambda(NC)$ = {19,23,28,32,38,45,50,55,60,71,81}. Accordingly, the unfolded state (λ_A) was defined by $\lambda_A = \lambda_0 \le 19$ and the folded state by taking $\lambda_B = \lambda_n = 81$. The number of trials per point at λ_i was $k_i = 10$ for $0 \le i < n$. Projected onto the free-energy landscape, Figure 1.11 also shows the λ response surface for the optimal order parameter obtained from the FFS-LSE method:

$$\lambda(NC, E) = -0.733 + 0.022(NC) - 0.004(E). \qquad (25)$$

Information of the folding mechanism can be obtained by investigating the specific amino acid residues and NC residues that contribute most significantly to the transition state. To this end, histograms for the frequency of each NC pair in the TSE were accumulated from the collection of configurations at the p_B contour of ½ ± σ [i.e., Eq. (25)] during a DFFS simulation. The standard error in the computed reaction coordinate model is $\sigma \approx 0.15$ (i.e., $MSE=\sigma^2$) and so the p_B = ½ contour includes values in the range [0.35, 0.65]. The initial region A was defined as $\lambda_A=\lambda_0=p_B(NC,E) = 0.0$ and the folded state region B as $\lambda_B=\lambda_n=p_B(NC,E)= 1$. The phase space was partitioned with eight interfaces (n=8) positioned at λ_i ($0 \le i \le$ n): $\lambda(NC,E)$ = {0.0, 0.1,0.2,0.3,0.4,0.5,0.6,0.85,1.0}.The number of trials at each λ_i was M_i=1000 for $0 \le i < n$ and statistics were accumulated for 100 blocks. The 13 amino acid pairs listed in Table 1.7 are the most probable native contacts that on average have a 90% chance or

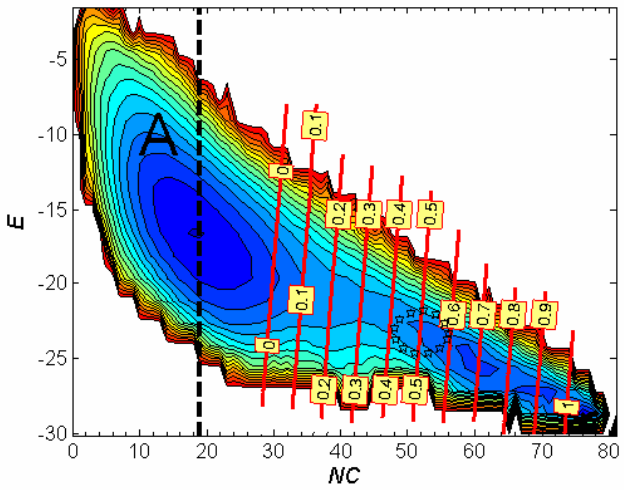

Figure 1.11 Estimated reaction coordinate iso-lines for the 64-mer lattice protein model folding at T_f=0.27. Thin dotted (red) lines are the predicted p_B committors from FFS-LSE, whose values appear as labels. Contour plot of the free energy landscape [energy (E) vs. nativeness (NC)] is shown only for visual reference. The color scheme changes from highest (red) to lowest (blue) elevations. The initial region is delimited by the dotted line labeled A. The most probable visited transition state region is bounded by the ellipse with center at (μ_{NC}=51, μ_E =-23.25).

more to occur in the TSE; they are listed in order of decreasing probability. These results are consistent with those obtained from the TSE found from a committor analysis[11] (data not shown). Figure 1.11 also shows the most populated intermediate region (given by the λ=0.5 iso-surface) as an ellipse with center at (μ_{NC}=51, μ_E =-23.25) and lengths of major and minor semiaxes σ_{NC} = 4 and σ_E = 1.42, respectively. Notice that the ellipse is a bit off center because of limitation of the FFS-LSE fit and the statistical error in the estimation of the p_B values. Figure 1.12 shows snapshots of typical folding trajectories for this protein; it can be observed that the 13 *NC* pairs with greater occurrence probability in the TSE (p_B≈0.5, Q~0.65) correspond to the nucleus that begins to form at early stages (p_B≈0.1 Q~0.2) during the folding process (the residues are shaded to display the formation of *NCs*). Note that the darker (red) residues are clustered in a patch, representing a core of NC pairs that is essential for the transition from unfolded to native state and contribute (on average) to 22.2% of the potential energy of the folded structure. This nucleation scenario is consistent with that previously reported in experimental[35,36] and simulation[3,37] studies. In general, the folding mechanism for the lattice protein begins with the peptide chain in a highly fluctuating unfolded state characterized by extended regions that persist up until the TS is reached. It then switches to a series of conformations with a partial, native-like structure, as indicated by the emergence of a persistent darker (red) patch (i.e., a nucleus of *NCs*). After that, there is a rapid transition from the partially folded intermediate state to more compact native-like conformations (at p_B≈0.8, Q~0.80). This transition corresponds to the collapse of fluctuating internal loops in the protein. In the final stage of folding, external loops at the surface of the protein rearrange to attain the folded structure.

Table 1.7 Most probable native contacts found in the transition state ensemble for 64-mer lattice protein at $T_f = 0.27$. They are listed in order of decreasing probability.

NC pairs (*i*,*j*)	
***i* (type)**	***j* (type)**
2(E)	35(K)
3(K)	24(D)
3(K)	26(D)
24(D)	37(K)
25(I)	28(L)
25(I)	36(L)
26(D)	35(K)
27(T)	30(G)
27(T)	34(A)
28(L)	33(I)
29(Q)	32(P)
31(S)	34(A)
33(I)	36(L)

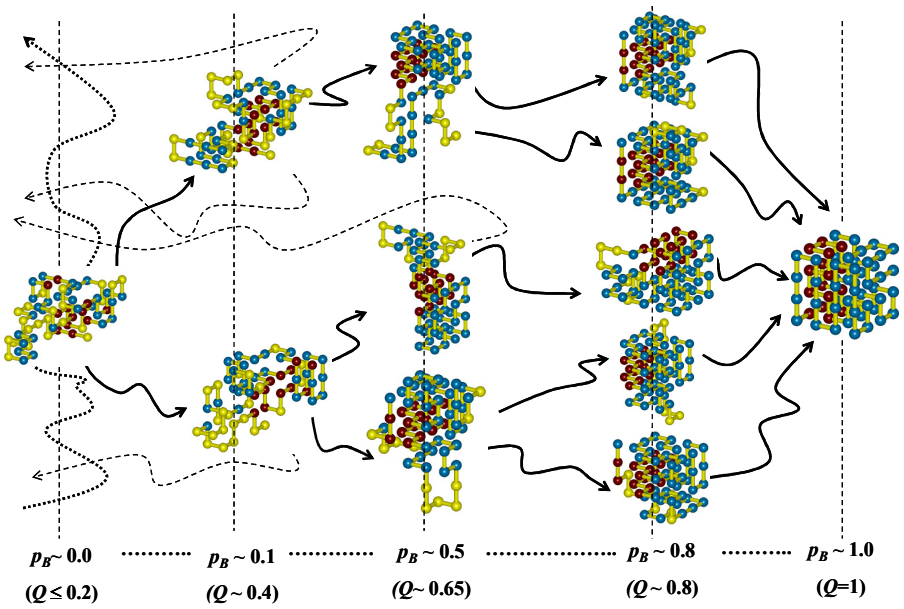

Figure 1.12 Representative snapshots for the folding event of the 64-mer lattice protein at T_f = 0.27. The simulation trajectory in the unfolded basin is shown by a thick dotted line. Trial runs (k_i) fired from λ_i that successfully reached the next λ_{i+1} interface are shown by a thick line and those which failed to reach λ_{i+1} are shown by a dotted line. Amino acid contacts are coded by the color of the residue: gray/blue indicates native contacts; whereas black/red designates the native contacts listed in Table 1.7, which have higher probability to belong to the transition state. The final frame shows the native state conformation.

V. Conclusion

How does one identify transition states and a good reaction coordinate for rare events? Answering this ubiquitous question implies finding an ensemble of reactive trajectories (TPE) which is most efficiently found from transition path sampling methods. FFS-type algorithms, for example, can generate transition paths that partially overcome the problem of reaction coordinate selection. However, some of the challenges for the application of these algorithms include the determination of an adequate order parameter to describe transition state regions, and the assessment of computational efficiency. In this article, we followed the idea that the best reaction coordinate to describe rare events, in fact the reaction coordinate, consists of the iso-committor p_B surfaces. This idea is not new, [9,11,19,21,22,24] but our approach differs from these previous approaches in that the path sampling is independent of the reaction coordinate to be tested. Thus, the proposed FFS-LSE approach better lends itself to applications in complex systems where the reaction coordinate cannot be identified a priori. The FFS-LSE method generates estimates for p_B "on-the-fly" from FFS-type simulations using an initial guess of the order parameter. The p_B history is then used to fit a model for the reaction coordinate in terms of several collective variables by LSE, and the significant terms in the model are determined by an analysis of variance. In this way, an optimized reaction coordinate is obtained that depends on a few relevant variables. The collective variables are evaluated at each stored point for which the p_B value is known, so that p_B is calculated only once for a set of states on the TPE. In this work, we used the BG algorithm to estimate the p_B values because in this scheme each point at any interface λ_i is sampled by firing a minimum number (k_i) of trial runs to the next interface, thus ensuring reasonably good statistics for the estimation of p_B values via Eq. (5). However, the DFFS scheme (rather than BG) could also be used to get p_B

estimates if one only considers points for which a minimum number of trials runs have been fired. In general, we envision that the dynamics of a system can be studied via two FFS-type simulations: the first one intended to determine a good reaction coordinate, and the second one intended to get mechanistic details of the process kinetics (using the good reaction coordinate found first). In the protein folding examples in this work, we used BG for the first stage and DFFS for the second.

The proposed method was applied to find reaction coordinates and transition states in three simple test examples. The examples of a particle moving on a bistable potential energy surface proved that the method allows the identification of the exact TS dividing surface from the p_B history. The example on the genetic switch shows that FFS-LSE can be applied to the estimation of reaction coordinates for nonequilibrium, rare-event problems. Finally, the example of the lattice protein folding shows that the FSS-LSE method leads to an improved order parameter (relative to other typical good choices) and thus, to a higher computational efficiency for FFS-type simulations. Moreover, we showed that the FFS-LSE approach provides a simple way in which information about the transition path ensemble (TPE), transition state ensemble (TSE) and hence mechanistic details of the dynamics of the system could be obtained while calculating the transition rate constant. Hence, once the TS dividing surface is known, the TSE can be extracted by simply screening the characteristics of the collective variables around the $p_B \sim ½$ contour during a single FFS-type simulation without the need to perform additional committor simulations.

While the simple examples studied here illustrate the validity and efficiency of the proposed method, in future work we plan to compare its performance with that of other variants, including the ML approach.[19] We argue, however, that the FFS-LSE approach should be seen as a complementary, rather than a competing approach to find good reaction coordinates (e.g., for systems with stochastic dynamics or non-

equilibrium states for which FFS may be more suitable than TPS). The FFS-LSE method could become a valuable tool for studying the dynamics in complex systems where the identification of a good reaction coordinate is essential for the efficient use of path sampling approaches.

Acknowledgments

EB is grateful to the Sloan Foundation for a graduate Fellowship. Additional support from the National Science Foundation Award BES-0093769 is acknowledged.

Appendix

The "Direct" Forward Flux path sampling algorithm (DFFS)

The DFFS method is similar to BG with the difference that in DFFS one samples M_i randomly selected points at each λ_i, rather that sampling k_i trials runs per stored point at λ_i. After an equilibration period, a first simulation is performed to store points each time the trajectory leaves A and crosses λ_0. This run in the basin A is suspended when N_0 points at λ_0 have been collected. In the second stage of the algorithm, partial paths are generated from this initial configurations stored at λ_0 to estimate the conditional probabilities, $P(\lambda_{i+1} \mid \lambda_i)$, of reaching λ_{i+1} from λ_1. Starting with the collection of N_0 points at λ_0, M_0 trials runs are fired from randomly selected such points, and are continued until either reaching λ_1 or returning to the initial region A. Then, each end-point configuration $N_S^{(0)}$ resulting from successful trial runs to reach λ_1 is stored and used as starting point for M_1 trial runs toward λ_2 (or back to A). If $N_S^{(1)}>0$ of the trial runs reach λ_2, the partial paths are continued by initiating M_2 trials runs to λ_3 from each of the $N_S^{(1)}$ successful configurations. This procedure is repeated using the collection of points generated at previous interfaces and firing trials that run as far as λ_{i+1} or λ_A, until either the final region $\lambda_n=\lambda_B$ is reached or because no

successful trials were generated at some intermediate interfaces. In this way, an estimated value of $P(\lambda_{i+1} \mid \lambda_i) = N_S^{(i)} / M_i$ can be obtained for each interface. As with the BG method, $P(\lambda_{n=B} \mid \lambda_0)$ is obtained by using Eq. (2) and the flux $\overline{\Phi}_{A,0} / \overline{h}_A$ in Eq. (1) is obtained from the simulation run in basin A by dividing the total number of crossing configurations at λ_0 by the total length of this run. For a complete description of the FFS path sampling scheme, the readers are referred to Ref. 2.

REFERENCES

[1] R. J. Allen, D. Frenkel, and P. R. ten Wolde, J. Chem. Phys. 124, 194111 (2006).

[2] R. J. Allen, D. Frenkel, and P. R. ten Wolde, J. Chem. Phys. 124, 024102 (2006).

[3] E. E. Borrero and F. A. Escobedo, J. Chem. Phys. 125, 164904 (2006).

[4] D. Chandler, *Introduction to Morden statistical Mechanics*. (Oxford University Press, New York, 1987).

[5] D. Frenkel and B. Smit, *Understanding Molecular Simulation: From Algorithms to Applications*, 2nd ed. (Academic, Boston, 2002).

[6] J. B. Anderson, Adv. Chem. Phys. 91, 381 (1995).

[7] G. E. Crooks and D. Chandler, Phys. Rev. E 6402, 026109 (2001).

[8] P. G. Bolhuis, Proc. Natl. Acad. Sci. U. S. A. 100, 12129 (2003).

[9] P. G. Bolhuis, D. Chandler, C. Dellago, and P. L. Geissler, Annu. Rev. Phys. Chem. 53, 291 (2002).

[10] C. Dellago, P. G. Bolhuis, F. S. Csajka, and D. Chandler, J. Chem. Phys. 108, 1964 (1998).

[11] C. Dellago, P. G. Bolhuis, and P. L. Geissler, Adv. Chem. Phys. 123, 1 (2002).

[12] A. K. Faradjian and R. Elber, J. Chem. Phys. 120, 10880 (2004).

[13] T. S. van Erp and P. G. Bolhuis, J. Comput. Phys. 205, 157 (2005).

[14] T. S. van Erp, D. Moroni, and P. G. Bolhuis, J. Chem. Phys. 118, 7762 (2003).

[15] D. Moroni, T. S. van Erp, and P. G. Bolhuis, Physica A (Amsterdam) 340, 395 (2004).

[16] D. Chandler, J. Chem. Phys. 68, 2959 (1978).

[17] T. S. van Erp, J. Chem. Phys. 125 (2006).

[18] R. J. Allen, P. B. Warren, and P. R. ten Wolde, Phys. Rev. Lett. 94, 018104 (2005).

[19] B. Peters and B. L. Trout, J. Chem. Phys. 125 (2006).

[20] R. B. Best and G. Hummer, Proc. Natl. Acad. Sci. U. S. A. 102, 6732 (2005).

[21] G. Hummer, J. Chem. Phys. 120, 516 (2004).

[22] A. Ma and A. R. Dinner, Protein Sci. 13, 219 (2004).

[23] L. Maragliano, A. Fischer, E. Vanden-Eijnden, and G. Ciccotti, J. Chem. Phys. 125 (2006).

[24] E. Weinan, W. Q. Ren, and E. Vanden-Eijnden, Chem. Phys. Lett. 413, 242 (2005).

[25] D. G. Mongomery, *Desing and Analysis of Experiments*, 5th ed. (John Wiley & sons, Inc., New York, 2001).

[26] I. J. Myung, J. Math. Psych. 47, 90 (2003).

[27] P. G. Bolhuis, www.science.uva.nl/~bolhuis/tps/content/exercise.pdf (2007).

[28] V. I. Abkevich, A. M. Gutin, and E. I. Shakhnovich, Fold Des. 13, 221 (1996).

[29] L. M. Contreras, F. Martinez-Veracoechea, P. Pohkarel, A. D. Stroock, F. Escobedo, and M. P. DeLisa, Biotechnol Bioeng 94, 105 (2006).

[30] B. Peters, J. Chem. Phys. 125 (2006).

[31] D. Bratko and H. W. Blanch, J. Chem. Phys. 114, 561 (2001).

[32] D. Bratko and H. W. Blanch, J. Chem. Phys. 118, 5185 (2003).

[33] K. Leonhard, J. M. Prausnitz, and C. J. Radke, Phys. Chem. Chem. Phys. 5, 5291 (2003).

[34] K. Leonhard, J. M. Prausnitz, and C. J. Radke, Biophysical Chemistry 106, 81 (2003).

[35] A. R. Fersht, Proc. Natl. Acad. Sci. U. S. A. 92, 10869 (1995).

[36] A. R. Fersht, Curr. Opin. Struct. Biol. 7, 3 (1997).

[37] V. S. Pande and D. S. Rokhsar, Proc. Natl. Acad. Sci. U. S. A. 96, 1273 (1999).

2. CHAPTER: Optimizing the sampling and staging for simulations of rare events via forward flux sampling schemes[*]

Abstract

In this work, we present an adaptive algorithm to optimize the phase space sampling for simulations of rare events in complex systems via forward flux sampling (FFS) schemes. In FFS, interfaces are used to partition the phase space along an order parameter λ connecting the initial and final regions of interest. Since the kinetic "bottleneck" regions along the order parameter are not usually known beforehand, an adaptive procedure is used that first finds these regions by estimating the rate constants associated with reaching subsequent interfaces; thereafter, the FFS simulation is reset to concentrate the sampling on those bottlenecks. The approach can optimize for either the number and position of the interfaces (i.e., optimized λ phase staging), and/or the number M of fired trial runs per interface (i.e., the $\{M_i\}$ set) to minimize the statistical error in the rate constant estimation per simulation period. For example, the optimization of the λ staging leads to a net constant flux of partial trajectories between interfaces and hence a constant flux of connected paths throughout the region between the two end states. The method is demonstrated for several test systems, including the folding of a lattice protein. It is shown that the proposed approach leads to an optimized λ staging and $\{M_i\}$ set which increase the computational efficiency of the sampling algorithm.

I. Introduction

Understanding the kinetics of complex stochastic events from molecular simulations is a long-standing goal in computational studies. Several path sampling

[*] E. E. Borrero and F. A. Escobedo, J. Chem. Phys. 129, 024115 (2008).

methods have been proposed to facilitate simulations of such rare events, like transition path sampling (TPS),[1-4] milestoning methods,[5] transition interface sampling (TIS),[6,7] partial path transition interface sampling (PPTIS)[8] and "forward flux sampling" (FFS)-type methods.[9,10] These sampling schemes require a choice of reaction coordinate (i.e., order parameter) to describe the system's dynamic. For example, TPS uses an order parameter to distinguish the two stable states in the free energy landscape to estimate the average transition rate constant. The transition path ensemble (TPE) is then sampled by using a Monte Carlo (MC) procedure that generates new paths by shooting forward and backward in time from old paths of fixed length, which limits its applicability to systems in equilibrium. Alternative path sampling methods, like TIS[7] and PPTIS[8] have been developed, which allow the sampling of variable path lengths and improve the efficiency of the TPE sampling via a series of interfaces along an order parameter that partition the phase space between two stable states. Milestoning methods[5] are very similar to PPTIS, but trajectories between interfaces are generated assuming a steady-state distribution at each interface (i.e., a strong Markovian condition), an assumption that limits their applicability. More recently, FFS-type algorithms have been proposed which, like TIS and PPTIS, allow the computation of both rate constants and TPE by dividing the phase space between the initial and final region into a series of interfaces. FFS relies on the generation of partial trajectories between interfaces, in which the crossing points at the next interface of successful trial runs are stored and used to initiate new partial pathways for the following interface.[9-11] The TPE is then obtained by connecting sequentially generated partial paths between the interfaces. However, the challenges entailed in applying interface-based methods like FFS to complex systems include: (i) the choice of an adequate order parameter that allows the description of the transition state regions and (ii) the choice of a suitable partitioning of the phase-space between

the stable states (i.e., number and position of the interfaces). [12] Recently, we addressed the first challenge by proposing a new algorithm for identifying suitable reaction coordinates to describe the progression of rare events in complex systems denoted as FFS-LSE.[13] The method is based on the FFS technique and standard least-square estimation (LSE). FFS-LSE uses the transition path ensemble (TPE) obtained from FFS to obtain "on-the-fly" estimates of the committor probability to the final region, p_B. LSE is used to identify an optimized order parameter model constructed with any number of candidate collective properties, which is able to correlate these p_B data. However, even though a good order parameter is obtained and then used to partition the phase-space, the efficiency of the sampling is still sensitive to the number and position of the interfaces and to how extensively different interfaces are sampled. In this paper, we address this challenge by applying an adaptive algorithm which reduces the statistical error of the estimated transition rate constant (for a given computational cost) by optimizing either (i) the number of trial runs per interface (the $\{M\}$ set) for fixed positions of the interfaces (the $\{\lambda\}$ set) or (ii) the position of the interfaces $\{\lambda\}$ for a given $\{M\}$ set.

Similar optimization strategies have been used previously in the literature for evaluating the free energy along an order parameter wherein, for example, it has been customary to assume that the most efficient sampling is achieved by allowing a uniform sampling of all intermediates and end states. This so-called "flat-histogram" [14,15] approach introduces "biasing" weights into the ensemble probability density function as a way to enhance the sampling in the space of interest using an iteration algorithm which simultaneously improves the estimation of free energies (i.e., weights) and the sampling. An alternative approach is that of Trebst et al.[16] who presented an algorithm which systematically optimizes the biasing weights to maximize the number of round trips (per CPU time) between the lower and upper

bounds of the sampling space of interest. Their approach uses an adaptive loop that reweighs the ensemble based on preceding measurements of local diffusivity to detect the "bottlenecks", and reallocates computational resources to such regions, generally resulting in a non-flat sampling histogram. This approach can minimize both the system's equilibration time and the statistical errors in the free energy over the sampled domain and has also been applied to improve the efficiency of parallel tempering methods to identify an optimal set of temperatures.[17] More recently, Escobedo et al.[18,19] proposed an algorithm to optimize the staging (i.e., the intermediate values of a physical variable used to regulate the coupling or decoupling of a target molecule during the insertion-deletion process) on expanded ensemble simulations. For the systems considered, they found that optimizing the staging was more effective than optimizing the sampling frequency of fixed stages for maximizing round trips and decreasing statistical errors.

For FFS-type simulations where the sampling is achieved by dividing the target phase space between the initial and final region into a series of interfaces, it is expected that by minimizing the error in the estimate of the conditional probabilities to reach subsequent interfaces (or the probability to reach the final region from the first interface), one should attain a reduced statistical error in the estimated transition rate constant for a fixed computational cost. It is reasonable then to assume that the most efficient path sampling could be achieved by concentrating the placement of interfaces in the region previous to the transition state (TS), i.e., while climbing over the free-energy barrier, where the vast majority of the trial runs per interface return to the initial region. However, since the free energy profile along the order parameter is not usually known beforehand, an iterative procedure is needed in which the complex free energy landscape is explored first to allow concentrating the sampling to the interfaces near the "bottlenecks" of the FFS simulation. Hence, our strategy is to simultaneously

improve the estimation of the conditional probabilities to reach subsequent interfaces (which yield information on "interfacial rate constants") and the sampling of the phase space. By minimizing the variance of the estimated transition rate constant (for fixed computational cost) we arrive at a recipe that effectively increases the net flux of partial trajectories between interfaces and hence the flux of connected paths everywhere between the two stable states.

By way of background, in Sec. II we start by reviewing briefly the formalism of FFS-type schemes for the calculation of rate constants and transition pathways. In Sec III, we introduced the basis and details of the proposed optimization algorithm. Finally, in Sec. IV the usefulness of the method is demonstrated for several test systems, including the case of a particle moving in a two-dimensional energy surface (Sec. IV A), a genetic switch (Sec. IV B), and a lattice protein model (Sec. IV C).

II. FFS-Type Sampling Algorithm

The rare paths (between stable or meta-stable states) in the FFS-type sampling scheme are captured by essentially performing a stochastic sampling of dynamic trajectories between interfaces. To this end, an order parameter $\lambda(x)$ is used to partition the phase space of interest such that the system has values $\lambda(x) \leq \lambda_A(x)$ in region A and $\lambda(x) \geq \lambda_B(x)$ in region B. Here, x denotes the coordinates of the phase space. The paths belonging to the transition path ensemble (TPE) are generated such that any trajectory from A to B passes through each interface in turn. The rate constant of the process is then estimated as an average rate of transitions using an "effective positive flux" expression.[9-11]

$$k_{A\to B} = \overline{\Phi}_{A,0} P\left(\lambda_{n=B} \mid \lambda_0\right) \qquad (1)$$

Equation (1) expresses the rate constant $k_{\mathrm{A}\to\mathrm{B}}$ for transitions from A to B as the product of a small flux from A to the first interface (λ_0), $\overline{\Phi}_{A,0}$, and the probability that

a trajectory reaching λ_0 from A will reach to B without returning to A, $P(\lambda_{n=B} \mid \lambda_0)$ [9]. Moreover, the use of interfaces to partitioning the phase space allows the determination of the $P(\lambda_{n=B} \mid \lambda_0)$ value as the product of conditional probabilities to reach subsequent interface:

$$P(\lambda_{n=B} \mid \lambda_0) = \prod_{i=0}^{n-1} P(\lambda_{i+1} \mid \lambda_i) \quad (2)$$

where $P(\lambda_{i+1} \mid \lambda_i)$ is the probability that a trajectory that visits A and crosses λ_i for the first time will subsequently reach λ_{i+1} without returning to the initial region A.[12] The statistics required to estimate these conditional probabilities are obtained from the TPE which in turn is obtained using, e.g., the "direct" Forward Flux Sampling (DFFS), the Branched Growth method (BG), or the Rosenbluth method (RB).[9,11] In this work, we focused on the DFFS and BG sampling schemes, which we implement as described below.

The first stage of the algorithm starts with a simulation in the basin of attraction of A. After an equilibration period, each time the trajectory leaves A and crosses λ_0 the phase space coordinates of the system at that point are stored. This run is then suspended once N_0 points at λ_0 have been collected, and the flux $\overline{\Phi}_{A,0}$ in Eq. (1) can be calculated as the total number of crossing configurations at λ_0 divided by the total length of this run. In the second stage of the algorithm, partial paths are generated from these initial configurations stored at λ_0 and the conditional probabilities, $P(\lambda_{i+1} \mid \lambda_i)$, of reaching λ_{i+1} from λ_i are estimated.

In the DFFS method, the collection of stored points at λ_0 are used to fire M_0 trials runs, which are continued until either reaching λ_1 or returning to the initial region A. Each trial run starts from a random point selected from the N_0 points at λ_0. Then, each end point configuration $N_S^{(0)}$ resulting from successful trial runs to reach λ_1

is stored and used as starting point for M_1 trial runs toward λ_2 (or back to A). If $N_S^{(1)}>0$ of the trial runs reach λ_2, the partial paths are continued by initiating M_2 trials runs to λ_3, randomly chosen from the $N_S^{(1)}$ successful configurations. This procedure is repeated until either the final region $\lambda_n=\lambda_B$ is reached or no successful trials were generated at some intermediate interfaces. The conditional probabilities in Eq. (2) are then calculated by $P(\lambda_{i+1} \mid \lambda_i) = N_S^{(i)} / M_i$ for each interface, and $P(\lambda_{n=B} \mid \lambda_0)$ is obtained by multiplying these together.

The BG method is similar to DFFS with the difference that in BG one samples k_i trial runs per stored point at λ_i, rather that sampling M_i randomly selected points at λ_i. Starting from a randomly selected configuration at λ_0, k_0 trial runs are fired and continued until either reaching λ_1 or returning to region A. Then, each end-point configuration $N_S^{(0)}$ resulting from successful trial runs reaching λ_1 is stored and used as starting point for k_1 trial runs toward λ_2 (or back to A). If $N_S^{(1)}>0$ of the trial runs reach λ_2, the partial paths are continued by initiating k_2 trials runs to λ_3 from each of the $N_S^{(1)}$ successful configurations. This procedure is repeated until either the final region $\lambda_n=\lambda_B$ is reached or no successful trials were generated at some intermediate interface. In this way, an estimate of $P(\lambda_{n=B} \mid \lambda_0)$ is obtained as the total number of branches that eventually reach λ_n, divided by the total possible number of branches, i.e., $P(\lambda_{n=B} \mid \lambda_0) = N_S^{(n-1)} / \prod_{i=0}^{n-1} k_i$. The final estimate of $P(\lambda_{n=B} \mid \lambda_0)$ is obtained by averaging the results over many such branching paths generated by following the same procedure outlined above starting from different randomly choosen points at λ_0.

For both DFFS and BG schemes, the TPE is obtained beginning with the collection of trials which arrive at $\lambda_B=\lambda_n$ from λ_{n-1} and tracing back the sequence of connected partial paths which link them to region A. For a complete description of the FFS path sampling schemes, the readers are referred to Ref. 10. It is was shown in Ref. [13] that estimates of the committor probability p_B for *individual* interfacial

points can be readily obtained using FFS-type methods. We now note that *average* values of $\langle\ p_B\ \rangle_\lambda$ for a given interface can also be readily estimated by using recursively the equation:

$$\left\langle p_B \right\rangle_{\lambda_i} = \left\langle p_B \right\rangle_{\lambda_{i+1}} P(\lambda_{i+1} \mid \lambda_i) \quad (3)$$

for i=n-1, n-2,…,1 with the boundary condition that $\left\langle p_B \right\rangle_{\lambda_n} = \left\langle p_B \right\rangle_{\lambda_B} = 1$. Note also that $\left\langle p_B \right\rangle$ could also be estimated from the more expensive analysis entailed by the standard committor analysis.[3,20] When the order parameter is a good estimate or the true reaction coordinate of the system and number of connecting paths is large enough, Eq. (3) will give a good approximation to the mean values of the committor distributions of different λ's.

III. Formulation of the adaptive optimization algorithm

The computational efficiency (ε) for FFS-type simulations is usually estimated as:

$$\varepsilon = 1/[C\nu] \quad (4)$$

where C represents the computational cost, and ν stands for the relative variance (i.e., statistical error) in the estimated value $k_{\mathrm{A}\to\mathrm{B}}$ of the rate constant per starting point at λ_0.[11] Following Eq. (1), the variance in the estimate of $k_{\mathrm{A}\to\mathrm{B}}$ depends on the relative variance of both $\overline{\Phi}_{A,0}$ and $P(\lambda_{n=B} \mid \lambda_0)$. The error in $\overline{\Phi}_{A,0}$ could be ignored as long as: (i) λ_0 is positioned close enough to the region A, and (ii) the simulation run in A is made long enough to estimate $\overline{\Phi}_{A,0}$ with high accuracy[11] and a computational cost that is minimal compared to that for estimating $P(\lambda_{n=B} \mid \lambda_0)$. Hence, treating $\overline{\Phi}_{A,0}$ as a constant, the variance, V, in $k_{\mathrm{A}\to\mathrm{B}}$ is given by:

$$V[k_{A\to B}] \propto V[P(\lambda_{n=B} \mid \lambda_o)]. \qquad (5)$$

The statistical error ν in $k_{\mathrm{A}\to\mathrm{B}}$ is defined to be $V[\,P(\lambda_{n=B} \mid \lambda_0)\,]$ per initial point at λ_0, divided by the square of the expectation value $\mathrm{E}[\,P(\lambda_{n=B} \mid \lambda_0)\,]$:[11]

$$\nu \equiv N_0 \frac{V\left[P(\lambda_n \mid \lambda_0)\right]}{P(\lambda_n \mid \lambda_0)^2} \qquad (6)$$

where N_0 is the number of starting points at λ_0 and $V\left[P(\lambda_n \mid \lambda_0)\right] = \overline{\left(P(\lambda_n \mid \lambda_0)\right)^2} - \left(\overline{P(\lambda_n \mid \lambda_0)}\right)^2$. This definition of ν allows us to analyze the efficiency of the order parameter used in the FFS-type simulations in a systematic way, only calculating the variance in the estimate of $P(\lambda_{n=B} \mid \lambda_0)$. Note that in the derivation of Eq. (6), we have assumed that the $N_S^{(i)}$ successful trials runs at different interfaces *i* are uncorrelated. In reality, of course, we could expect correlation between interfaces, especially if the interfaces are closely spaced or the system dynamics have a large degree of "memory". However, we expect this assumption to have a minor impact in situations where one employs a modest number of interfaces.

The computational cost [factor C in Eq. (4)] is defined to be the average number of simulation steps required by a particular FFS-type sampling scheme per starting point at λ_0. If the average cost of generating one starting point at λ_0 (i.e., the cost of estimate the flux $\overline{\Phi}_{A,0}$ from region A to λ_0) is ignored as well as any other contributions to the CPU time, such as memory storage; the cost C_i of firing one trial run from interface λ_i can be approximated by [11]

$$C_i = S\left\{P(\lambda_{i+1} \mid \lambda_i)\left[\lambda_{i+1} - \lambda_i\right] + \left(1 - P(\lambda_{i+1} \mid \lambda_i)\right)\left[\lambda_i - \lambda_A\right]\right\}. \quad (7)$$

where the average length of a partial trajectory from λ_i to λ_j is assumed to be linearly proportional to $|\lambda_j - \lambda_i|$, with a proportionality constant S; and each fired run is continued until it reaches either the next interface λ_{i+1} [with probability $P(\lambda_{i+1} \mid \lambda_i)$] or the boundary λ_A of region A [with probability (1- $P(\lambda_{i+1} \mid \lambda_i)$)].[11] Based on these assumptions, Allen et al.[11] derived expressions for C for each FFS-type sampling scheme, taking into account the possibility that the cost is reduced by failing to reach later interfaces (i.e., as in the case that M_i or k_i is small).

Our formulation for optimized sampling of the λ phase space focuses on the reduction of the statistical error in the estimate of $k_{A\to B}$, which in turn results in the minimization of the variance in the $P(\lambda_{n=B} \mid \lambda_0)$ estimate [see Eq. (6)]. Following Allen et al.[11], $V[\,P(\lambda_{n=B} \mid \lambda_0)\,]$ is expressed in terms of the relative variances $V[\,P(\lambda_{i+1} \mid \lambda_i)\,]$ through the use of Eq. (2):

$$V\left[P(\lambda_n \mid \lambda_0)\right] = \sum_{i=0}^{n-1}\left(\frac{\partial P(\lambda_n \mid \lambda_0)}{\partial P(\lambda_{i+1} \mid \lambda_i)}\right)^2 V\left[P(\lambda_{i+1} \mid \lambda_i)\right] \approx P(\lambda_n \mid \lambda_0)^2 \sum_{i=0}^{n-1} \frac{V\left[P(\lambda_{i+1} \mid \lambda_i)\right]}{P(\lambda_{i+1} \mid \lambda_i)^2} \qquad (8)$$

In Eq. (8), we have made use of the basic error propagation formula:[11]

$$V\left[f(x,y,...)\right] = \left(\frac{\partial f}{\partial x}\right)^2 V[x] + \left(\frac{\partial f}{\partial y}\right)^2 V[y] + \ldots, \qquad (9)$$

where $f(x,y,\ldots)$ is a function of multiple uncorrelated variables x, y,… and the partial derivatives are evaluated with all variables at their mean values. In the next sections, we give expressions for $V[\,P(\lambda_{i+1} \mid \lambda_i)\,]$ for the DFFS and BG schemes and use them to increase the sampling efficiency.

A. Optimizing the efficiency of the DFFS scheme

Starting with the collection of N_0 stored states at λ_0, $P(\lambda_{i+1} \mid \lambda_i)$ is obtained by firing M_i trial runs for each interface: $P(\lambda_{i+1} \mid \lambda_i) = N_s^{(i)} / M_i$, where $N_s^{(i)}$ is the number of trials which reach λ_{i+1}. Using the fact that M_i is a constant during the simulation, we can express the variance $V[P(\lambda_{i+1} \mid \lambda_i)]$ in the estimated values of $P(\lambda_{i+1} \mid \lambda_i)$ as:

$$V[P(\lambda_{i+1} \mid \lambda_i)] = \frac{V[N_s^{(i)}]}{M_i^2} \ . \qquad (10)$$

Here, we have used again Eq. (9). Furthermore, the events being counted $N_s^{(i)}$ are actually the outcomes of M_i discrete trials fired from interface λ_i, which can be modeled using the binomial distribution,[11] with mean

$$E\left[N_s^{(i)}\right] = M_i P(\lambda_{i+1} \mid \lambda_i) \qquad (11)$$

and variance

$$V\left[N_s^{(i)}\right] = M_i P(\lambda_{i+1} \mid \lambda_i)\left(1 - P(\lambda_{i+1} \mid \lambda_i)\right). \qquad (12)$$

Finally, using Eq. (8), (10), and (12), we find that:

$$V^{DFFS}\left[P(\lambda_n \mid \lambda_0)\right] = P(\lambda_n \mid \lambda_0)^2 \sum_{i=0}^{n-1} \frac{1 - P(\lambda_{i+1} \mid \lambda_i)}{M_i P(\lambda_{i+1} \mid \lambda_i)} \qquad (13)$$

and from Eq. (6)

$$\nu^{DFFS} \equiv N_0 \sum_{i=0}^{n-1} \frac{1 - P(\lambda_{i+1} \mid \lambda_i)}{M_i P(\lambda_{i+1} \mid \lambda_i)}. \qquad (14)$$

For decreasing the statistical error in the estimated value $k_{A\to B}$ and thus increase the efficiency of the sampling, we consider here two approaches:

Approach 1: For fixed λ staging, optimize the values of trial runs M_i at each interface for a given computational cost.

Approach 2: For fixed M_i values (so that the computational cost is relatively constrained), optimize the staging of the λ phase space.

1. Approach 1: Optimizing the {M_i} set

The statistical error in the estimate of $k_{A\to B}$, or equivalently the variance in $P(\lambda_{n=B} \mid \lambda_0)$ is reduced by minimizing Eq. (13) with the constraint that the computational cost

$$C^{DFFS} = R + \frac{1}{N_0}\left\{ M_0 C_0 + \sum_{i=1}^{n-1}\left[M_i C_i \prod_{j=0}^{i-1}\left[1-\left(1-P(\lambda_{j+1}\mid\lambda_j)\right)^{M_j}\right]\right]\right\} \quad (15)$$

must remain constant. In Eq. (15), R is the average cost of generating one starting point at λ_0, which does not depend in the λ staging and the $\{M_i\}$ set, and $\prod_{j=0}^{i-1}\left[1-\left(1-P(\lambda_{j+1}\mid\lambda_j)\right)^{M_j}\right]$ is the probability that at least one trial run at i>0 is successful at all interfaces j<i, which corrects the cost C_i (of firing one trial run from interface λ_i) by the possibility that none of the trials runs from λ_i reach λ_{i+1} and the DFFS procedure is terminated at interface i (thus reducing C^{DFFS}).[11] Since, R and S are constants, Eq. (15) can be rewritten as:

$$Cost^{DFFS} = (C^{DFFS} - R)\frac{N_0}{S} = M_0\frac{C_0}{S} + \sum_{i=1}^{n-1}\left[M_i \frac{C_i}{S}\prod_{j=0}^{i-1}\left[1-\left(1-P(\lambda_{j+1}\mid\lambda_j)\right)^{M_j}\right]\right] \quad (16)$$

where $Cost^{DFFS}$ is a constant. The function to minimize is then Eq. (13) plus a Lagrange multiplier (α) to restrict $Cost^{DFFS}$:

$$\Im = P(\lambda_n \mid \lambda_0)^2 \sum_{i=0}^{n-1} \frac{1-P(\lambda_{i+1} \mid \lambda_i)}{M_i P(\lambda_{i+1} \mid \lambda_i)} + \alpha\left(M_0 \frac{C_0}{S} + \sum_{i=1}^{n-1}\left[M_i \frac{C_i}{S} \prod_{j=0}^{i-1}\left[1-\left(1-P(\lambda_{j+1} \mid \lambda_j)\right)^{M_j}\right]\right] - Cost^{DFFS} \right) . \quad (17)$$

Varying the distribution of M_i values at λ_i, Eq. (17) is minimized (i.e., $\partial\Im / \partial M_i = 0$) for

$$M_i = \frac{P(\lambda_n \mid \lambda_0)}{\sqrt{\alpha}} \left(\frac{1-P(\lambda_{i+1} \mid \lambda_i)}{P(\lambda_{i+1} \mid \lambda_i)} \right)^{1/2} \times \left(\begin{array}{l} \dfrac{C_i}{S} \displaystyle\prod_{j=0}^{i-1}\left[1-\left(1-P(\lambda_{j+1} \mid \lambda_j)\right)^{M_j}\right] \\ - M_i \left(1-P(\lambda_{j+1} \mid \lambda_j)\right)^{M_j} \displaystyle\sum_{k=i+1}^{n-1}\left[M_k \frac{C_k}{S} \prod_{\substack{j=0, \\ j\neq i}}^{k-1}\left[1-\left(1-P(\lambda_{j+1} \mid \lambda_j)\right)^{M_j}\right]\right] \end{array} \right)^{-1/2} . \quad (18)$$

Now, if we assume that M_i is large enough that at least one trial run reaches λ_{i+1}, so that $\left[1-P(\lambda_{i+1} \mid \lambda_i)\right]^{M_i} \approx 0$, Eqs. (15) and (18) are simplified, respectively, to:

$$C^{DFFS} = R + \frac{1}{N_0} \sum_{i=0}^{n-1} M_i C_i \quad (19)$$

$$M_i = \frac{P(\lambda_n \mid \lambda_0)}{\sqrt{\alpha}} \left(\frac{1-P(\lambda_{i+1} \mid \lambda_i)}{P(\lambda_{i+1} \mid \lambda_i)} \right)^{1/2} \left(P(\lambda_{i+1} \mid \lambda_i)\left[\lambda_{i+1} - \lambda_i\right] + \left(1-P(\lambda_{i+1} \mid \lambda_i)\right)\left[\lambda_i - \lambda_A\right] \right)^{-1/2} . \quad (20)$$

To implement Eq. (20), we can first set M_i of a chosen interface i to a desired value M (i.e., $M_0 = M$), and find the other M_i values such that:

$$\frac{M_{i+1}}{M_i} = \left(\frac{1 - P(\lambda_{i+2} \mid \lambda_{i+1})}{1 - P(\lambda_{i+1} \mid \lambda_i)} \frac{P(\lambda_{i+1} \mid \lambda_i)}{P(\lambda_{i+2} \mid \lambda_{i+1})} \right)^{1/2} \times$$
$$\left(\frac{P(\lambda_{i+1} \mid \lambda_i)[\lambda_{i+1} - \lambda_i] + (1 - P(\lambda_{i+1} \mid \lambda_i))[\lambda_i - \lambda_A]}{P(\lambda_{i+2} \mid \lambda_{i+1})[\lambda_{i+2} - \lambda_{i+1}] + (1 - P(\lambda_{i+2} \mid \lambda_{i+1}))[\lambda_{i+1} - \lambda_A]} \right)^{1/2}. \quad (21)$$

Hence, the chosen value of M sets the computational cost of the DFFS simulation. These results [from Eq. (21)] can be used as an initial trial solution to iteratively solve Eq. (18).

Now, if we assumed that $P(\lambda_{i+1} \mid \lambda_i)$ is small, so that $[1 - P(\lambda_{i+1} \mid \lambda_i)] >> P(\lambda_{i+1} \mid \lambda_i)$, Eq. (20) is simplified to:

$$\frac{M_{i+1}}{M_i} = \left(\frac{P(\lambda_{i+1} \mid \lambda_i)[\lambda_i - \lambda_A]}{P(\lambda_{i+2} \mid \lambda_{i+1})[\lambda_{i+1} - \lambda_A]} \right)^{1/2}. \quad (22)$$

In Sec. IV A, we compare the results for the statistical error in the estimate of $k_{A \to B}$ obtained for simulations with optimized $\{M_i'\}$ sets determined from either Eqs. (18) and (22).

2. Approach 2: Optimizing the {λ} set (staging).

For approach 2, the variance in $P(\lambda_{n=B} \mid \lambda_0)$ is reduced by minimizing Eq. (13) with the constraint that $P(\lambda_{n=B} \mid \lambda_0) = \prod_{i=0}^{n-1} P(\lambda_{i+1} \mid \lambda_i)$ [i.e., Eq. (2)] must remain constant. We do this by adding a Lagrange multiplier (α), and varying the probability distribution $P(\lambda_{i+1} \mid \lambda_i)$ such that

$$\frac{\partial \left\{ V^{DFFS}[P(\lambda_n \mid \lambda_0)] + \alpha P(\lambda_n \mid \lambda_0) \right\}}{\partial P(\lambda_{i+1} \mid \lambda_i)} = -\frac{P(\lambda_n \mid \lambda_0)^2}{M_i P(\lambda_{i+1} \mid \lambda_i)^2} + \alpha \frac{P(\lambda_n \mid \lambda_0)}{P(\lambda_{i+1} \mid \lambda_i)} = 0, \quad (23)$$

which leads to

$$M_i P(\lambda_{i+1} \mid \lambda_i) = P(\lambda_{n=B} \mid \lambda_0) / \alpha = N_s^{(i)} = N_s = \text{constant}. \quad (24)$$

Thus, for optimal sampling, the $P(\lambda_{i+1} \mid \lambda_i)$ values [which are determined by the $\{\lambda\}$ set] must be set to attain a net constant flux of partial trajectories between interfaces $N_s^{(i)} = N_s$ and hence a constant flux of connected paths between the two stable states per simulation period, N_s, this is illustrated in Figure 2.1. Note that Eq. (24) does not fully specify the $P(\lambda_{i+1} \mid \lambda_i)$ values since we could simultaneously change the $P(\lambda_{i+1} \mid \lambda_i)$ and M_i values to satisfy it. This freedom allows us to externally input a desirable distribution of $P(\lambda_{i+1} \mid \lambda_i)$ values, e.g., a uniform distribution with

$$P(\lambda_{i+1} \mid \lambda_i) = \left[P(\lambda_n \mid \lambda_0)\right]^{1/n}. \qquad (25)$$

Note that in this approach 2, we assumed that by fixing the M_i values (i.e., M_i=M) the computational cost C^{DFFS} could be sufficiently constrained. An alternative optimization approach for $\{\lambda'\}$ would entail adding an additional Lagrange multiplier to constraint the cost [i.e., Eq. (16)]. However, the analytical and numerical treatment of such a case is cumbersome and will not be discussed here. Note that one could use approaches 1 and 2 in turn to achieve a combined optimization effect; i.e., to optimize first the staging (approach 2) for a specific prescribed set of $P(\lambda_{i+1} \mid \lambda_i)$ values and then optimize for the $\{M_i\}$ set (approach 1), as will be shown in Sec. IV A and C. Note also that Eq. (24) provides a basis to estimate an optimized $\{M_i\}$ set for a particular $\{\lambda\}$ set (and thus a $P(\lambda_{i+1} \mid \lambda_i)$ distribution) that could be seen as an alternative to using Eq. (18) or (22). In Sec. IV A, we will precisely compare the statistical error in $k_{A\to B}$ associated with such methods.

To implement approach 2, we also need an algorithm that will allow us to go from the optimized or desired probability $P(\lambda_{i+1} \mid \lambda_i)$ distribution to the corresponding $\{\lambda_i'\}$ set. This procedure is explained in detail in Sec. III C.

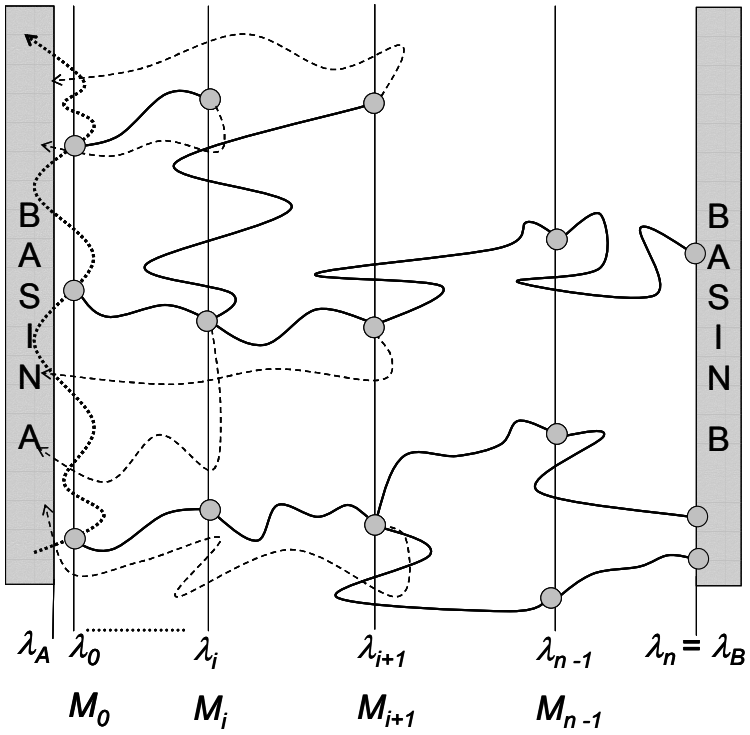

Figure 2.1 . A schematic view of the generation of branched paths (thick lines) using the direct forward flux sampling method (DFFS). The first stage is the simulation run in the A basin shown by a dotted line. Starting points for the subsequent generation of branched paths are labeled with a gray circle at λ_0. The second stage correspond to the trial runs (M_i) fired from λ_i ; those that reached the next λ_{i+1} interface are shown by a thick line and those which failed to reach λ_{i+1} are shown by a dotted line. In this example, the staging of the λ phase space was optimized [i.e., by varying $\{M_i\}$ set (i.e., $M_i = N_s / P(\lambda_{i+1} \mid \lambda_i)$) or by relocating the interface positions such that $P(\lambda_{i+1} \mid \lambda_i)$ is constant and a net flux of three partial trajectories between interfaces per simulation period was obtained.

B. Optimizing the efficiency of the BG scheme

Allen et al.[11] have determined the expressions for the variance V^{BG} in the estimated value of $P(\lambda_{n=B} \mid \lambda_0)$ and the statistical error ν^{BG}, which, we present below:

$$V^{BG}\left[P(\lambda_n \mid \lambda_0)\right] = \frac{P(\lambda_n \mid \lambda_0)^2}{N_0} \sum_{i=0}^{n-1} \frac{1 - P(\lambda_{i+1} \mid \lambda_i)}{\prod_{j=0}^{i} k_j P(\lambda_{j+1} \mid \lambda_j)} \qquad (26)$$

and

$$\nu^{BG} \equiv \sum_{i=0}^{n-1} \frac{1 - P(\lambda_{i+1} \mid \lambda_i)}{\prod_{j=0}^{i} k_j P(\lambda_{j+1} \mid \lambda_j)} \qquad (27)$$

where Eq. (26) has been divided by N_0 to account for the fact that $P(\lambda_{n=B} \mid \lambda_0)$ is calculated by averaging results over N_0 starting points at λ_0.

1. Approach 1: Optimizing the {k_i} set

To decrease ν, we can optimize the values of trial runs k_i at each interface for a fixed computational cost C^{BG}. Note that in this approach the staging is maintained fixed to the initial $\{\lambda_i\}$ set. The optimized $\{k_i'\}$ set is then found by choosing the k_i values such that Eq. (26) is minimized with the constraint that

$$C^{BG} = R + k_0 C_0 + \sum_{i=0}^{n-1} \left[k_i C_i \prod_{j=0}^{i-1} P(\lambda_{j+1} \mid \lambda_j) k_j \right] \qquad (28)$$

must remain constant. In Eq. (28), R is the average cost of generating one starting point at λ_0, which does not depend on the λ staging or the $\{k_i\}$ set.[11] Then, using Eq. (7) and the fact that R and S are constants, Eq. (28) can be rewritten as:

$$Cost^{BG} = \frac{(C^{BG} - R)}{S} = k_0 \{P(\lambda_1 \mid \lambda_0)[\lambda_1 - \lambda_0] + (1 - P(\lambda_1 \mid \lambda_0))[\lambda_0 - \lambda_A]\}$$
$$+ \sum_{i=0}^{n-1} \left[k_i \{P(\lambda_{i+1} \mid \lambda_i)[\lambda_{i+1} - \lambda_i] + (1 - P(\lambda_{i+1} \mid \lambda_i))[\lambda_i - \lambda_A]\} \prod_{j=0}^{i-1} P(\lambda_{j+1} \mid \lambda_j) k_j \right] \quad (29)$$

where $Cost^{BG}$ is a constant. Hence, Eq. (26) is minimized by adding a Lagrange multiplier (α) to constrain the $Cost^{BG}$ in Eq. (29), and varying the distribution of k_i values such that

$$\frac{\partial \left\{ V^{BG}[P(\lambda_n \mid \lambda_0)] + \alpha Cost^{BG} \right\}}{\partial k_i} = -\frac{P(\lambda_n \mid \lambda_0)^2}{N_0} \frac{1}{k_i^2 P(\lambda_{i+1} \mid \lambda_i)} \sum_{m=i}^{n-1} \frac{1 - P(\lambda_{m+1} \mid \lambda_m)}{\prod_{\substack{l=0, \\ l \neq i}}^{m} k_l P(\lambda_{l+1} \mid \lambda_l)} +$$
$$\alpha \left\{ \frac{C_i}{S} \prod_{l=0}^{i-1} k_l P(\lambda_{l+1} \mid \lambda_l) \right\} + \alpha \left\{ P(\lambda_{i+1} \mid \lambda_i) \sum_{m=i+1}^{n-1} \left[k_m \frac{C_m}{S} \prod_{\substack{l=0, \\ l \neq i}}^{m-1} k_l P(\lambda_{l+1} \mid \lambda_l) \right] \right\} = 0 \quad (30)$$

which leads to

$$k_i = \frac{P(\lambda_n \mid \lambda_0)}{\sqrt{\alpha N_0}} \left(\frac{1}{P(\lambda_{i+1} \mid \lambda_i)} \sum_{m=i}^{n-1} \frac{1 - P(\lambda_{m+1} \mid \lambda_m)}{\prod_{\substack{l=0, \\ l \neq i}}^{m} k_l P(\lambda_{l+1} \mid \lambda_l)} \right)^{1/2} \times$$
$$\left[\frac{C_i}{S} \prod_{l=0}^{i-1} k_l P(\lambda_{l+1} \mid \lambda_l) + P(\lambda_{i+1} \mid \lambda_i) \sum_{m=i+1}^{n-1} \left[k_m \frac{C_m}{S} \prod_{\substack{l=0, \\ l \neq i}}^{m-1} k_l P(\lambda_{l+1} \mid \lambda_l) \right] \right]^{-1/2} . \quad (31)$$

2. Approach 2: Optimizing the {λ} set (staging)

In the BG method, a branching tree of paths connecting A to B is generated from a single point at λ_0 and, unlike the situation in the DFFS method, now the

number of trials runs fired at each interface M_i depends on the $P(\lambda_{i+1} \mid \lambda_i)$ values. Hence, for a given number $N_s^{(i)}$ of points at λ_i, the total number of trials fired is $N_s^{(i)} k_i$, i.e.:

$$M_i = k_i \prod_{j=0}^{i-1} k_j P(\lambda_{j+1} \mid \lambda_j) \,. \qquad (32)$$

Assuming that C^{BG} is sufficiently constrained by fixing all M_i values (as we did for the DFFS, Sec. III A) leads to:

$$\begin{aligned} k_0 &= M_0 && i = 0 \\ k_i &= \frac{M_i / M_{i-1}}{P(\lambda_{i+1} \mid \lambda_i)} && 0 < i < n \end{aligned} \,. \qquad (33)$$

To minimize the variance in $P(\lambda_{n=B} \mid \lambda_0)$, Eq. (26) is rewritten on account of Eq. (33) and form the function $\Im$ to enact the constraint $P(\lambda_{n=B} \mid \lambda_0) = \prod_{i=0}^{n-1} P(\lambda_{i+1} \mid \lambda_i)$ [i.e., Eq. (2)] via a Lagrange multiplier (α):

$$\Im = \frac{P(\lambda_n \mid \lambda_0)^2}{N_0} \sum_{i=0}^{n-1} \frac{1 - P(\lambda_{i+1} \mid \lambda_i)}{M_0 P(\lambda_1 \mid \lambda_0) \prod_{j=1}^{i} \frac{M_j P(\lambda_{j+1} \mid \lambda_j)}{M_{j-1} P(\lambda_j \mid \lambda_{j-1})}} + \alpha \left(\prod_{i=0}^{n-1} P(\lambda_{i+1} \mid \lambda_i) - P(\lambda_n \mid \lambda_0) \right) \qquad (34)$$

To which the extremum condition is applied by varying the distribution of $P(\lambda_{i+1} \mid \lambda_i)$:

$$\frac{\partial \Im}{\partial P(\lambda_{i+1} \mid \lambda_i)} = -\frac{1}{N_0} \frac{P(\lambda_n \mid \lambda_0)^2}{M_i P(\lambda_{i+1} \mid \lambda_i)^2} + \alpha \frac{P(\lambda_n \mid \lambda_0)}{P(\lambda_{i+1} \mid \lambda_i)} = 0 \,, \qquad (35)$$

This results in the same expression obtained for the DFFS scheme [i.e., Eq. (24)], $M_i P(\lambda_{i+1} \mid \lambda_i) = P(\lambda_{n=B} \mid \lambda_0) / (\alpha N_0) = N_s^{(i)} = N_s$ = constant. As with the DFFS method,

we have some freedom to specify the $P(\lambda_{i+1} \mid \lambda_i)$ distribution, but for $M_i P(\lambda_{i+1} \mid \lambda_i)$ to remain constant, k_i has to be also chosen such that

$$k_0 = N_s / P(\lambda_1 \mid \lambda_0) \qquad i = 0$$
$$k_i = \frac{N_s / P(\lambda_{i+1} \mid \lambda_i)}{\prod_{j=0}^{i-1} k_j P(\lambda_{j+1} \mid \lambda_j)} = \frac{1}{P(\lambda_{i+1} \mid \lambda_i)} \qquad 0 < i < n \qquad (36)$$

where we have used Eqs. (24) and (32). Note that $k_i = 1/P(\lambda_{i+1} \mid \lambda_i)$ for $0<i<n$ while k_0 is chosen so that $N_s = k_0 P(\lambda_1 \mid \lambda_0)$ is fixed to the desired number of partial paths between interfaces (i.e., number of trajectories per starting point at λ_0).

C. Getting a new {λ} staging

The purpose of this calculation is essentially to "interpolate" from the current $P(\lambda_{i+1} \mid \lambda_i)$ vs. λ data to find a new $\{\lambda'\}$ set for which the $P(\lambda_{i+1} \mid \lambda_i)$ values are distributed according to a prescribed criterion; i.e., according to specified values of $\{P(\lambda_1|\lambda_0)$, $P(\lambda_2|\lambda_1)$, …$\}$ [e.g., as in Eq. (25) wherein $P(\lambda_{i+1}|\lambda_i)$ = constant]. To do this, we need to generate a function f of $P(\lambda_{i+1} \mid \lambda_i)$ that when plotted as a function of λ, it will be invariant (independent on the λ values or their number) and will provide a one-to-one correspondence between an f value and a λ value. Such a f function will be the bridge to go from any prescribed $P(\lambda_{i+1} \mid \lambda_i)$ values to the sought-after λ values. For this purpose, a possibly natural choice for the f function is a normalized rate constant for processes going from λ_A to λ_i:

$$f_1(\lambda_i) = \frac{k(\lambda_A \to \lambda_i)}{k(\lambda_A \to \lambda_B)} = \frac{\prod_{j=0}^{i-1} P(\lambda_{j+1} \mid \lambda_j)}{\prod_{j=0}^{n-1} P(\lambda_{j+1} \mid \lambda_j)}, \qquad i = 1,...,n-1 \qquad (37)$$

where $k(\lambda_A \rightarrow \lambda_B)=k_{A\rightarrow B}$ is simply a constant. Note that this f_1 function is a monotonically decreasing function of λ (going from a large value for λ_1 to a value of unity for $\lambda_n=\lambda_B$), so that for any given f_1 value, there exists a unique λ value. From a numerical interpolation perspective, however, Eq. (37) is not very convenient since $f_1(\lambda_1)$ is unbound and f_1 could span several orders of magnitude. A better function f can be constructed by noting that to a first approximation (small $\Delta\lambda = \lambda_{i+1}-\lambda_i$), one could expect that:

$$P(\lambda_{i+1} \mid \lambda_i) \approx \exp[-(\lambda_{i+1} - \lambda_i)/\overline{\lambda}_i] = \exp[-\Delta\lambda/\overline{\lambda}_i] \tag{38}$$

where $\overline{\lambda}_i$ is a positive quantity that only depends on the value of λ_i, and could be interpreted as the mean distance traveled forward by trajectories started at λ_i. Equation (38) satisfies the boundary conditions for P; i.e., $P(\lambda_{i+1} \mid \lambda_i) = 1$ if $\Delta\lambda=0$ and $P(\lambda_{i+1} \mid \lambda_i) \rightarrow 0$ if $\Delta\lambda\rightarrow\infty$, and suggests a monotonic one-to-one relationship between $\ln\left[P(\lambda_{i+1} \mid \lambda_i)\right]$ and $\Delta\lambda$. A more suitable f function would then be:

$$f(\lambda_i) = \frac{\ln[k(\lambda_A \rightarrow \lambda_i)]}{\ln[k(\lambda_A \rightarrow \lambda_B)]} = \frac{\sum_{j=0}^{i-1} \ln P(\lambda_{j+1} \mid \lambda_j)}{\sum_{j=0}^{n-1} \ln P(\lambda_{j+1} \mid \lambda_j)}, \qquad i = 1,...,n \tag{39}$$

Noting that according to Eqs. (38) and (39) $df/d\lambda \approx [f(\lambda_{i+1})-f(\lambda_i)]/\Delta\lambda \propto [-\ln P(\lambda_{i+1}|\lambda_i)]/\Delta\lambda, \propto \overline{\lambda}_i^{-1}$ and so f could be taken to represent a "cumulative" probability distribution corresponding to the probability density function $[-\ln P(\lambda_{i+1}|\lambda_i)]/(\lambda_{i+1}-\lambda_i)$ (interpretable as the reciprocal of the mean forward distance $\overline{\lambda}_i$). Function f of Eq. (39) monotonically increases with λ, going from $f(\lambda_0)=0$ to $f(\lambda_n=\lambda_B)=1$. Note that while the choice of f function is not unique, in this work we exclusively use Eq. (39). Since this f function should be independent of the discrete λ values used to generate it (in the initial simulation), one can then reverse the process so that given a set of

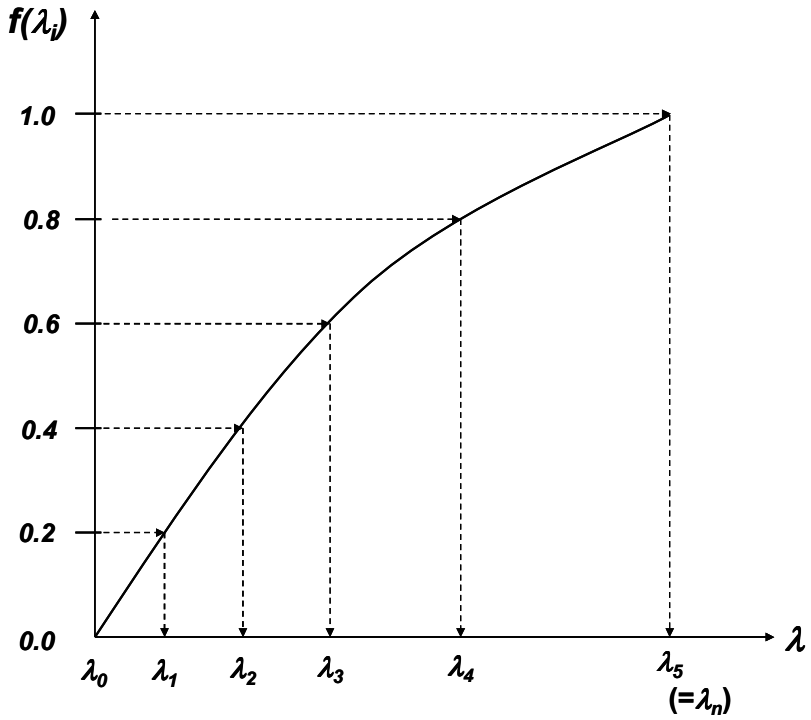

Figure 2.2 Schematics of the selection of a new {λ'} set from a measured $P(\lambda_{i+1} \mid \lambda_i)$ distribution. $f(\lambda_i)$ is the function corresponding to Eq. (39). In this example, n=6 interfaces, and if we wanted to have a constant $P(\lambda_{i+1} \mid \lambda_i)$ value for all interfaces, we would implement Eq. (40) by selecting four equidistant points along the $f(\lambda_i')$ axis (i.e., in jumps of 0.2) and project them over the λ axis (as illustrated by the dashed arrows) to get the sought after λ_i' values.

$\{P(\lambda_1|\lambda_0)$, $P(\lambda_2|\lambda_1)$, …$\}$ values, one first calculates the corresponding f values [from Eq. (39)] and then uses these to obtain the corresponding λ values from the f vs. λ curve. Figure 2.2 illustrates this idea. This process (of running FFS to get P's for given λ's and then constructing the f function to find λ's for prescribed P's) can be iterated more than once to reaching full convergence of the optimized $\{\lambda'\}$ set. For the particular case that it is desired to have $P(\lambda_{i+1}|\lambda_i)$ = constant for i=1,2,..,n-1 [as per Eq. (25)], Eq. (39) reduces to

$$f(\lambda_i) = \frac{\sum_{j=0}^{i-1} \ln P(\lambda_{j+1} \mid \lambda_j)}{\sum_{j=0}^{n-1} \ln P(\lambda_{j+1} \mid \lambda_j)} = \frac{i}{n}, \qquad i = 1,2,...,n \qquad (40)$$

where $1<j<n$ (n+1 is the number of interfaces) and the two extremal $\lambda_0'=\lambda_0$ and $\lambda_n'=\lambda_n$ remain fixed. Equation (40) prescribes that the intermediate λ interfaces are to be distributed in such a way that $\Delta f = f(\lambda_{i+1})-f(\lambda_i)=1/n$ is constant.

IV. Applications

A. Potential energy surface

The first system studied involved a particle moving on a two-dimensional potential energy surface. We chose this simple system since it is very well understood and has been used previously to test TPS methods.[3,13] A contour graph of this energy landscape is illustrated in Figure 2.3, showing two well defined minima at (-1, 0) and (1, 0). The initial and final regions were defined by circles of radius 0.2 centered at the minima. The kinetics of the system was simulated using MC dynamics at $\beta = 1/k_B T = 10$. The move set entailed the sampling in each dimension of a normal distribution centered on the current point:

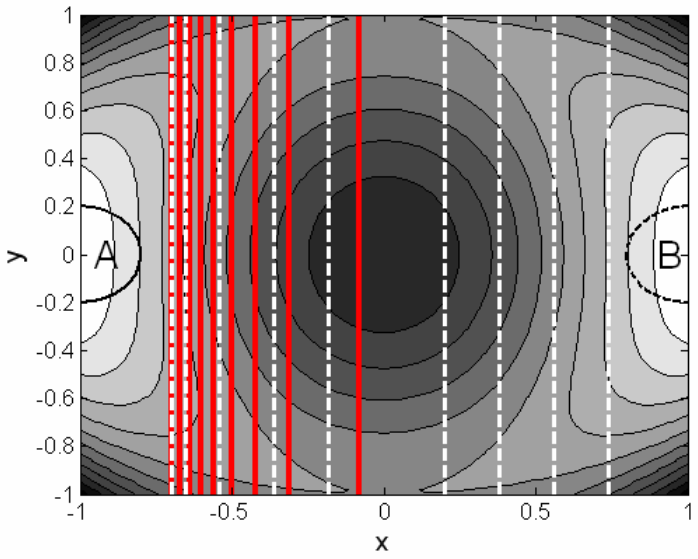

Figure 2.3 Contour graph of the free-energy surface for the two dimensional potential: $V(x,y)=\frac{1}{6}\left[4\left(1-x^2-y^2\right)^2+2\left(x^2-2\right)^2+\left(\left(x+y\right)^2-1\right)^2+\left(\left(x-y\right)^2-1\right)^2-2\right]$. The color scheme changes from highest (black) to lowest (white) elevations. The initial and final regions are shown by the circles labeled *A* and *B*, respectively. The initial λ staging for the FFS-type simulation is shown by dotted lines (white). The thick line (red) shows the optimized $\{\lambda_i'\}$ staging.

$$\eta(x_\alpha, x_{\alpha'}) = \frac{1}{\sigma\sqrt{2\pi}} e^{\left[-(x_\alpha - x_{\alpha'})^2 / 2\sigma^2\right]} \qquad (41)$$

where σ=0.04 is the standard deviation, chosen according to the distance the particle is expected to travel due to diffusion. Figure 2.3 also illustrates the initial λ staging surface (dotted lines): $\lambda_i = x$ for $0 \le i < n$. Accordingly, we defined the initial region by taking $\lambda_A = x \le$ -0.80 and the final region by taking $\lambda_B = \lambda_n \ge 0.9$. We used initially nine interfaces to partition the phase space (n=9), which are listed in Table 2.1. The initial number of trials per point at λ_i for the BG method was $k_0 = 200$ (i=0) and $k_i = 3$ ($1 \le i < n$). For DFFS, we choose the initial number M of trials per point at λ_i by taking $M_i = 220$ for $0 \le i < n$. The calculations were carried out as a series of blocks, each consisting of $N_0 = 100$ starting points at λ_0 for the BG method, and of one FFS run for the DFFS scheme. Results were averaged over ten blocks.

Table 2.1 shows the results of the proposed "approach 1" for the optimization of the $\{M_i'\}$ and $\{k_i'\}$ set using the DFFS and BG schemes, respectively; for the same initial λ staging. The optimized $\{M_i'\}$ was found by fixing M_0'=1000 and the M_i' values such that Eq. (18), (22) or (24) is satisfied for $1 \le i < n$. The optimized $\{k_i\}$ set was found by choosing k_0'=200 and the k_i' values such that Eq. (31) is satisfied for $1 \le i < n$. The M_0' and k_0' values were chosen so that the computational cost C match those from simulations with "unoptimized" $\{M_i'\}$ and $\{k_i'\}$ sets, respectively. C per starting points at λ_0 was estimated by using Eqs. (15) and (31) for the DFFS and BG scheme, respectively. Table 2.1 shows excellent agreement between both path sampling schemes: the values of $P(\lambda_{i+1} \mid \lambda_i)$ at each λ_i were found to be approximately the same.

While the reported optimized $\{M_i'\}$ and $\{k_i\}$ sets were obtained after three iterations, convergence is essentially attained after the first iteration.

Table 2.1 Optimized $\{M_i'\}$ and $\{k_i'\}$ sets, and values of $P(\lambda_{i+1} \mid \lambda_i)$ for the two-dimensional energy surface. The number of trial runs at λ_0 was fixed: M_0 = 1000 for the DFFS method, and $k_0 = 200$ ($i = 0$) for the BG scheme. For the case where Eq. (24) was used to obtain the optimized $\{M_i'\}$ set, the number of partial paths between interfaces $N_s^{(i)}$ was fixed to $\sim N_s = 250$. Hence, 250 trajectories connecting A to B were obtained per starting point at λ_0 for one DFFS run. Results were averaged over ten blocks.

Initial staging				**BG**		**DFFS**			
						M_i'			
i	λ_i	k_i	M_i	k_i'	$P(\lambda_{i+1}\|\lambda_i)$	**Eq. (18)**	**Eq. (22)**	**Eq. (24)**	$P(\lambda_{i+1}\|\lambda_i)$
0	-0.70	200	220	200	0.25	1000	1000	1000	0.25
1	-0.65	3	220	5	0.12	1214	1179	2083	0.12
2	-0.54	3	220	6	0.15	787	776	1563	0.16
3	-0.36	3	220	3	0.43	332	356	556	0.45
4	-0.18	3	220	2	0.48	237	282	490	0.51
5	0.20	3	220	2	0.82	140	175	305	0.82
6	0.38	3	220	2	0.89	108	154	278	0.90
7	0.56	3	220	1	0.96	73	139	260	0.96
8	0.74	3	220	1	0.99	41	129	253	1.00
9 (=n)	0.90	-	-	-	-	-	-	-	-

Table 2.2 shows the results of "approach 2" for the probability values of $P(\lambda_{i+1} \mid \lambda_i)$ and the optimized $\{\lambda_i\text{'}\}$ set from the iterative simulations using the DFFS and the BG schemes. The optimized λ staging reported was obtained after 3 iterations wherein $P(\lambda_{i+1} \mid \lambda_i) = \left[P(\lambda_n \mid \lambda_0)\right]^{1/n}$ was fixed as in Eq. (25). In Figure 2.4, the function $f(\lambda_i)$ [i.e., Eq (39)] for each λ_i interface is plotted versus (a) the actual λ values, and (b) the interface index (i.e., λ stage) for the DFFS method starting with the initial λ distribution listed in Table2.1. Figure 1.4(a) proves the uniqueness of $f(\lambda_i)$ and Figure 2.4(b) shows that convergence is attained after the first iteration. Since $M_i P(\lambda_{i+1} \mid \lambda_i)$ remains constant [i.e., Eq. (25)], M_i and k_i were also varied, so that the number of partial paths between interfaces $N_s^{(i)}$ was fixed to $\sim N_s$ =100 and 9 for DFFS and BG scheme, respectively. Hence, 9 and 100 trajectories connecting A to B were obtained per starting point at λ_0 for the BG method [i.e., $k_0 \approx 22$ (i=0) and $k_i \approx 3$ ($1 \le i < n$)], and for one DFFS run [i.e., $M_i \approx 230$ ($0 \le i < n$)], respectively . It is clear that both path sampling methods optimized the λ staging by partitioning the phase space with the same optimized $\{\lambda_i\text{'}\}$ set, so that $P(\lambda_{i+1} \mid \lambda_i) \approx 0.44$ for all 9 interfaces. The optimal $\{\lambda_i\text{'}\}$ staging obtained from the DFFS adaptive algorithm is illustrated in Figure 2.3 (thick lines). Note that (for a preset number of interfaces; i.e., n=9) the most efficient sampling is achieved by concentrating the placement of interfaces in the region preceding the transition state (TS), i.e., in the "bottlenecks" of the simulation. This can be seen in the contour graph of the free-energy surface associated with the transition; the TS is visually identifiable at $\lambda = x = 0$ (i.e., $p_B = 0.5$) where x stands for the x-coordinate and p_B for the committor probability to the final region. We expected $\lambda_{n\text{-}1}$ to be located before $\lambda = x = 0$ because approach 2 optimized for $P(\lambda_{n=B} \mid \lambda_{n-1}) = \langle p_B \rangle_{\lambda_{n-1}} \approx 0.44$ at $\lambda_{n\text{-}1}$, which is close to the committor probability

Table 2.2 Optimized $\{\lambda_i'\}$ set, and values of $P(\lambda_{i+1} \mid \lambda_i)$ for the two-dimensional energy surface. $M_i \approx 230$ $(0 \le i < n)$ for the DFFS method, and $k_0 \approx 22$ $(i = 0)$ and $k_i \approx 3$ $(1 \le i < n)$ for the BG scheme, so that the number of partial paths between interfaces $N_s^{(i)}$ was fixed to $\sim N_s$ =100 and 9, respectively. Hence, 9 and 100 trajectories connecting A to B were obtained per starting points at λ_0 for the BG method, and for one DFFS run, respectively. Results were averaged over ten blocks. The optimized set was found after three iterations.

i	**Initial {λ} set**	***Optimized* λ staging**			
		BG		**DFFS**	
	λ_i	λ_i'	$P(\lambda_{i+1}\|\lambda_i)$	λ_i'	$P(\lambda_{i+1}\|\lambda_i)$
0	-0.70	-0.70	0.442	-0.70	0.437
1	-0.65	-0.66	0.446	-0.67	0.438
2	-0.54	-0.64	0.439	-0.64	0.432
3	-0.36	-0.59	0.435	-0.60	0.439
4	-0.18	-0.55	0.441	-0.56	0.447
5	0.20	-0.50	0.437	-0.50	0.435
6	0.38	-0.42	0.435	-0.42	0.439
7	0.56	-0.30	0.439	-0.31	0.435
8	0.74	-0.07	0.446	-0.08	0.450
9 (=n)	0.90	0.90	-	0.90	-

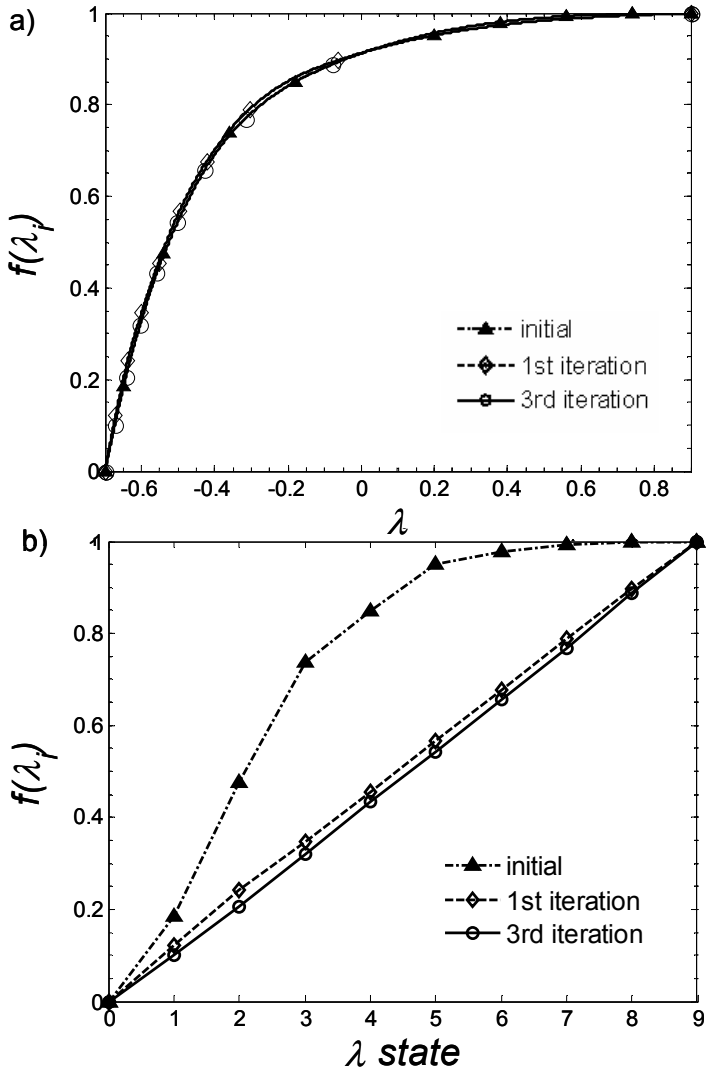

Figure 2.4 Function $f(\lambda_i)$ of Eq. (39) from DFFS simulations of the 2D potential system where λ ranges from λ_0 = -0.70 to $\lambda_{n=B}$ = 0.85. It shows $f(\lambda_i)$ as a function of a) the actual λ values, and b) the λ interface index. In part (b), notice that the choice of Eq. (25) leads to an optimized $\{\lambda'\}$ set (3rd iteration) where $\Delta f_i = f(\lambda_i) - f(\lambda_{i+1}) = 1/n$ (i.e., a constant slope).

value at the TSE (i.e., $p_B \approx 0.5$). Note that, according to Eq. (3), for the interface located at λ_{n-1}, the $P(\lambda_{n=B} \mid \lambda_{n-1}) = \langle p_B \rangle_{\lambda_{n-1}}$ value denotes the average value of the committor probability to the final region for states in the vicinity of $\lambda_{n-1} + \Delta\lambda$ where $\Delta\lambda << (\lambda_{n=B} - \lambda_{n-1})$.

In order to test that the optimized parameters increase the computational efficiency of the FFS-type simulations, we measure the statistical error (ν) in the estimated value $k_{A\to B}$ of the average transition rate constant as function of the CPU time. For each test case, simulations were carried out in a series of blocks. For DFFS, a block consists of a complete FFS calculation with N_0 starting points at λ_0. For the BG scheme, a block consists of $N_0 = 100$ starting points at λ_0. Each block produces an estimate of $P(\lambda_{n=B} \mid \lambda_0)$. To find $V[P(\lambda_n \mid \lambda_0)]$, we calculated the variance between blocks: $V[P(\lambda_n \mid \lambda_0)] = \overline{(P(\lambda_n \mid \lambda_0))^2} - \left(\overline{P(\lambda_n \mid \lambda_0)}\right)^2$, where the overbar denotes an average over the blocks. The computational cost C per starting point at λ_0 is the average CPU time per block, divided by N_0. All the runs were carried out in the same machine, a dual-core Opteron 270 (64 bit, and 2 GHz clock-speed). Note that these CPU time values will vary depending on the machine used and the coding details, however, the absolute values for the differences in the statistical error ($\Delta\nu$) is machine independent and should approach a particular plateau value.

Figure 2.5 shows the relative plateau value of ν (statistical error in $k_{A\to B}$) from DFFS simulations using the same initial λ staging. To approximately have the same computational cost C^{DFFS} [calculated by Eq. (15)], the ν values for the optimized $\{M_i'\}$ set determined by Eqs. (18), (22), and (24) (see Table 2.1) were compared to simulations with constant $M_i = M$ values [i.e., "unoptimized" $\{M_i\}$ set] of 340, 373, and 672, respectively. The relative error in ν was defined as the difference between the plateau values divided by error in the "unoptimized" simulation: $(\nu_{un-optimized} - \nu_{optimized})/\nu_{un-optimized}$. We observed a significant reduction in the statistical

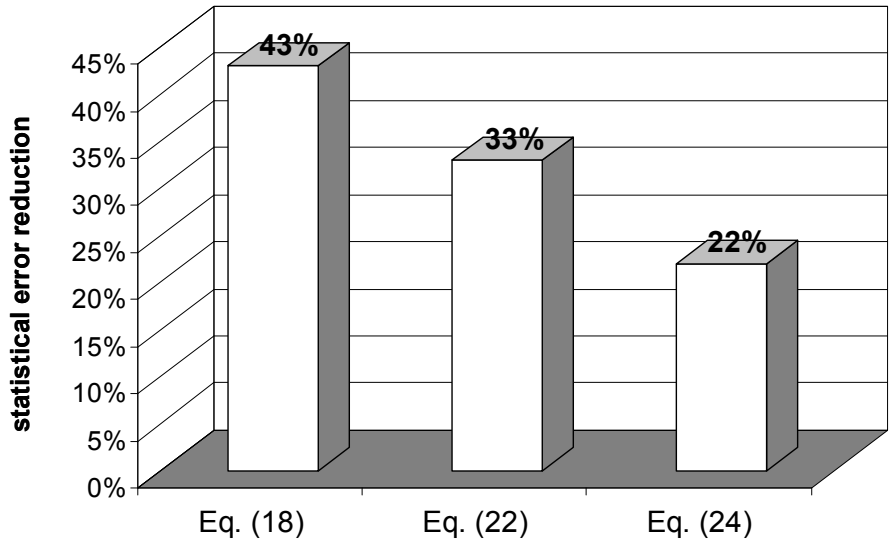

Figure 2.5 Relative statistical error ν (past the first 25 CPU time) in $k_{A\to B}$ for DFFS simulations for the 2D potential system. The minimum ν plateau values for simulations with optimized $\{M_i'\}$ sets determined by Eqs. (18), (22), and (24) were compared to those for simulations with "unoptimized" constant $M_i = M$ values having approximately the same computational cost C^{DFFS}.

error [43, 33, and 22% for optimized {M'} sets from Eqs. (18), (22), and (24), respectively], verifying that approach 1 leads to {M'} sets that improve the computational efficiency of the DFFS simulations. As expected, we observed a larger reduction (~ 43%) in the ν for the optimized {M_i'} set determined by Eq. (18) than those from the optimized {M_i'} set determined by Eq. (22) and (24) [i.e., simplified versions of Eq. (18)]. In Table 2.1, note that Eq. (22) estimates M_i values for the first interfaces (i.e., previous to the TSE) that are similar to those from Eq. (18), where it is accurate the assumption that $\left[1 - P(\lambda_{i+1} \mid \lambda_i)\right] >> P(\lambda_{i+1} \mid \lambda_i)$; however, it estimates M_i values ~two times greater than Eq. (18) for the last two interfaces; i.e., i=7 and 8, where $\left[1 - P(\lambda_{i+1} \mid \lambda_i)\right] << P(\lambda_{i+1} \mid \lambda_i)$. On the other hand, Eq. (24) overestimates the M_i values for all interfaces and leads to a reduction in ν that is half of that achieved by Eq. (18).

Figure 2.6 shows a comparison of the ν obtained with BG simulations for three cases: initial staging with un-optimized {k'} set, initial staging with optimized {k'} set, and optimized staging (plotted as a function of CPU time). For all three cases, we aimed to have approximately the same computational cost per starting points at λ_0 [estimated by Eq. (31)]; for this purpose, the number of partial paths between interfaces was fixed to $N_s^{(i)} = N_s = 9$ per starting points at λ_0 for the simulations with the optimized {λ_i'} set (listed in Table II). We observed a significant reduction in the statistical error [$\left(\nu_{un-optimized} - \nu_{optimized}\right)/\nu_{un-optimized} \approx 0.65$ and 0.76 for simulations using the initial staging with optimized {k'} set, and the optimized {λ'}, respectively], verifying again that our approach can tune the {λ_i'} set or the {k'} set to improve the computational efficiency of the BG simulations. Moreover, we observed that ν converged faster (to its plateau value) in CPU time for the optimized sets than for the initial unoptimized simulations. In other words, for the same CPU time, the BG

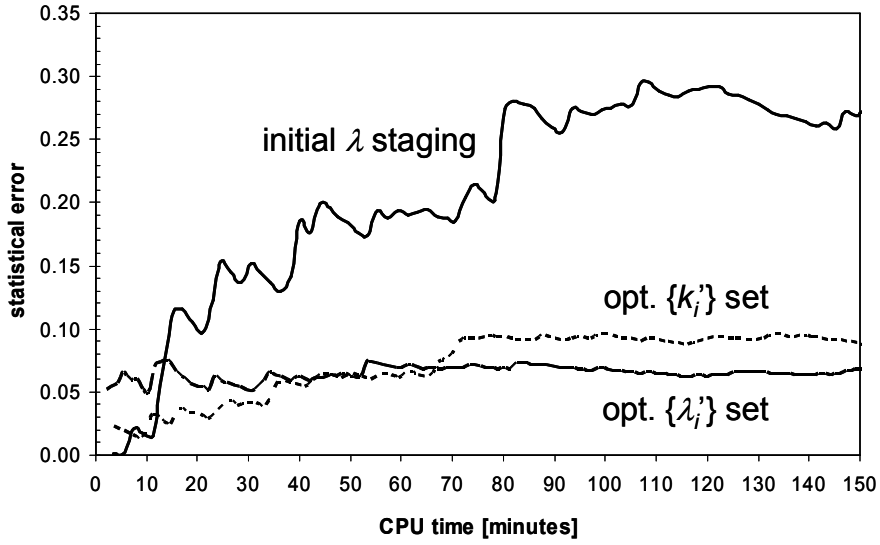

Figure 2.6 Results from BG simulations for the 2D potential system. Statistical error in the estimated $k_{A\to B}$ constant versus CPU time for: — initial staging with un-optimized $\{k'\}$ set, ------ initial staging with optimized $\{k'\}$ set, and — – — optimized $\{\lambda'\}$ set.

simulations with the optimized {k'} or {λ'} set required fewer blocks (i.e., fewer $P(\lambda_{n=B} \mid \lambda_0)$ estimates) to obtain a small error.

We now compare the plateau ν values (past the first 25 CPU minutes) obtained with DFFS (for 9 interfaces) for three cases: (i) initial staging with un-optimized {M} set, (ii) optimized staging, and (iii) optimized {λ'} (approach 2) followed by optimized {M'} (approach 1). For case (ii), the number of partial paths between interfaces was fixed to $N_s^{(i)} = N_s = 100$ per starting point at λ_0 such that the computational cost per starting points at λ_0 [estimated by Eq. (15)] equals the cost for simulations using a non-optimized λ staging. For case (iii), the optimized {M_i'} set [for the optimized λ staging of case (ii)] was found by fixing M_0'= 470 and the M_i' values such that Eq. (18) is satisfied for $1 \le i < n$; the M_0' value was chosen so that the computational cost C^{DFFS} per starting points at λ_0 matches the C^{DFFS} values from cases (i) and (ii) (see Table 2.2). The case (iii) simulations lead to a remarkable reduction of the statistical error (~73%) compared to those with the unoptimized {λ_i'} and {M_i'} sets ($\nu \approx 0.55$), converging to a plateau value ($\nu \approx 0.15$) that is less than half that for the case when only the {λ_i'} set was optimized ($\nu \approx 0.36$). These results show that the optimization approaches 1 and 2 can advantageously be used in combination.

In implementing approach 2 thus far, the {λ_i} was optimized by fixing a constant net flux of partial trajectories between interfaces and making the choice that $P(\lambda_{i+1} \mid \lambda_i)$ is constant [Eq. (25)]. But rather than fixing n (number of interfaces) and then computing $P(\lambda_{i+1} \mid \lambda_i)$, we can specify $P(\lambda_{i+1} \mid \lambda_i)$ and calculate the value of n needed. We thus performed a second series of DFFS optimization simulations for $P(\lambda_{i+1} \mid \lambda_i) = \left[P(\lambda_n \mid \lambda_0)\right]^{1/n}$ = 0.2, 0.3, 0.4, and 0.5, starting from the initial λ staging (n=9) listed in Table 2.3. Table 2.3 also shows the values of $P(\lambda_{i+1} \mid \lambda_i)$ at each λ_i and the optimized {λ_i'} set obtained after approximately three iterations with n = 4, 6, 8, and 11 for $P(\lambda_{i+1} \mid \lambda_i)$ = 0.2, 0.3, 0.4, and 0.5, respectively. Figure 2.7 also shows

Table 2.3 Optimized $\{\lambda_i'\}$ set, and values of $P(\lambda_{i+1} \mid \lambda_i)$ for the two-dimensional energy surface. The number of interfaces (n) and the λ staging was varied in order to obtain a desired constant $P(\lambda_{i+1} \mid \lambda_i)$ value by fixing $P(\lambda_{i+1} \mid \lambda_i) = \left[P(\lambda_n \mid \lambda_0)\right]^{1/n}$. M_i was also fixed so that $N_s^{(i)} = N_s = 100$. Hence, 100 trajectories connecting A to B were obtained per DFFS run (block). Results were averaged over ten blocks. The optimized set was found after three iterations.

i	**Initial {λ} set**	***Optimized* λ staging**							
		$[P(\lambda_n\|\lambda_0)]^{1/n} \approx 0.2$ [n=4]		**$[P(\lambda_n\|\lambda_0)]^{1/n} \approx 0.3$ [n=6]**		**$[P(\lambda_n\|\lambda_0)]^{1/n} \approx 0.4$ [n=8]**		**$[P(\lambda_n\|\lambda_0)]^{1/n} \approx 0.5$ [n=11]**	
	λ_i	λ_i'	$P(\lambda_{i+1}\|\lambda_i)$	λ_i'	$P(\lambda_{i+1}\|\lambda_i)$	λ_i'	$P(\lambda_{i+1}\|\lambda_i)$	λ_i'	$P(\lambda_{i+1}\|\lambda_i)$
0	-0.70	-0.70	0.167	-0.70	0.305	-0.70	0.396	-0.70	0.524
1	-0.65	-0.63	0.152	-0.65	0.286	-0.67	0.400	-0.68	0.504
2	-0.54	-0.52	0.169	-0.60	0.309	-0.63	0.405	-0.65	0.517
3	-0.36	-0.33	0.170	-0.53	0.303	-0.59	0.400	-0.62	0.526
4	-0.18	0.85 (=n)	-	-0.42	0.276	-0.53	0.393	-0.58	0.521
5	0.20	-	-	-0.20	0.309	-045	0.403	-0.55	0.519
6	0.38	-	-	0.90 (=n)	-	-0.34	0.398	-0.50	0.511
7	0.56	-	-	-	-	-0.11	0.402	-0.45	0.515
8	0.74	-	-	-	-	0.90 (=n)	-	-0.37	0.512
9	0.90 (=n)	-	-	-	-	-	-	-0.25	0.512
10	-	-	-	-	-	-	-	0.00	0.517
11	-	-	-	-	-	-	-	0.90 (=n)	-

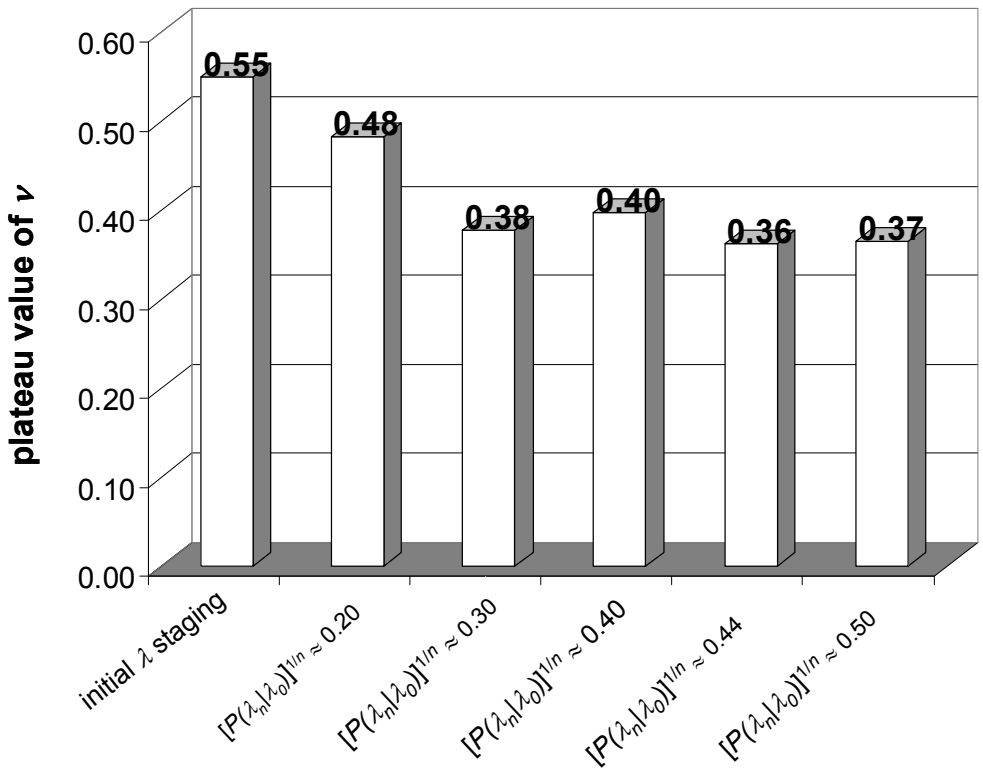

Figure 2.7 Plateau values for the statistical error (past the first 25 CPU time) in $k_{A\to B}$ for DFFS simulations of the 2D potential system. In these simulations the number of interfaces (n) and the λ staging was varied in order to obtain a desired constant $P(\lambda_{i+1} \mid \lambda_i)$ value by fixing $P(\lambda_{i+1} \mid \lambda_i) = \left[P(\lambda_n \mid \lambda_0)\right]^{1/n}$, as shown in Table 2.2 and 2.3.

the minimum statistical error ν (past the first 25 CPU time) in the estimated value of $k_{A\to B}$ for each one of these optimized $\{\lambda_i\text{'}\}$ set. As before, the same number of partial paths between interfaces $N_s^{(i)} = N_s = 100$ per starting points at λ_0 was fixed by taking M_i = 500, 333, 250, and 200 for $P(\lambda_{i+1} \mid \lambda_i)$ = 0.2, 0.3, 0.4, and 0.5, respectively. A significant reduction of the statistical error was again achieved, with plateau values of ν being smaller than that for the initial λ staging simulations ($\nu \approx 0.55$). However, we observed that the number of interfaces used to partition the λ phase space affects the achievable ν. For n= 4 (i.e., $\left[P(\lambda_n \mid \lambda_0)\right]^{1/n} = 0.2$), an intermediate ν plateau value of ~0.48 was reached, for n=6-8 $\nu \sim 0.38$, while for $n \geq 9$ ν converges to a minimum of ~0.36 (including data not shown for n=14 and n=21). Moreover, when $P(\lambda_{i+1} \mid \lambda_i)$ was fixed to $\left[P(\lambda_n \mid \lambda_0)\right]^{1/n} = 0.5$, Eq. (3) shows that the $P(\lambda_{n=B} \mid \lambda_{n-1})$ value at $\lambda_{n\text{-}1}$ corresponds to the average committor probability to the final region at the TSE, $\langle p_B \rangle_{\lambda_{n-1}}$, so that the method automatically locates $\lambda_{n\text{-}1}$ at $\lambda = x = 0$, as we expected and seen in Table 2.3.

Whether the reduction of statistical errors brought about by the optimization of $\{k_i\text{'}\}$ or $\{M_i\text{'}\}$ will be larger or smaller than that for the optimization of $\{\lambda_i\text{'}\}$ will strongly depend on the initial choice of these parameters (e.g., how "unoptimal" the λ values are) and on the number of interfaces. Figure 2.8 shows, for example, that ν converges faster to a smaller plateau value using an optimized $\{\lambda_i\text{'}\}$ set for simulations where the number of interfaces was fixed to be n=4. In these simulations, we used the BG method because it has been shown to be sensitive to changes in the n and k parameters.[11] The initial $\{\lambda_i\}$ set was: {-0.70, -0.60, -0.28, 0.17, 0.90} for $0 \leq \lambda_i \leq n$=4, k_0 = 200 and k_i = 7 ($1 \leq i < n$) which corresponds to $P(\lambda_{i+1} \mid \lambda_i)$ = {0.09, 0.03, 0.34, 0.70} for $0 \leq \lambda_i < n$=4. For approach 1, the optimized $\{k_i\text{'}\}$ set was {200, 13, 5, 2} to satisfy Eq. (31) for $1 \leq i < n$ and to match the computational cost C^{BG} of the initial $\{\lambda\}$, unoptimized $\{k_i\text{'}\}$ simulations via the k_0'=200 value. For approach 2, the

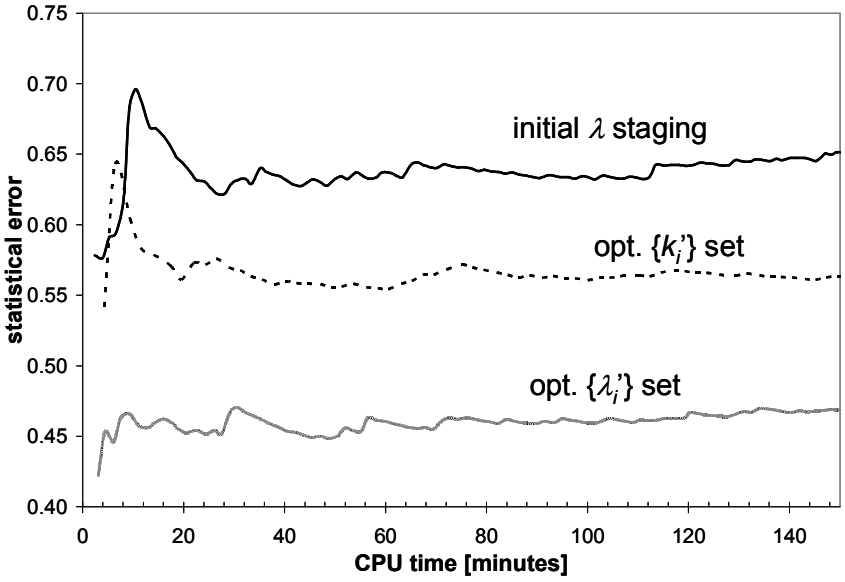

Figure 2.8 . Statistical error in the estimated $k_{A\to B}$ value versus CPU time for BG simulations of the 2D potential system for: — initial staging with un-optimized $\{k'\}$ set, ------ initial staging with optimized $\{k'\}$ set, and ◦◦◦ optimized $\{\lambda'\}$ set. In these simulations, the λ phase space was partitioned with four interfaces (n=4).

optimized $\{\lambda_i\text{'}\}$ set for $[P(\lambda_n|\lambda_0)]^{1/n} \approx 0.2$ and n=4 is listed in Table 2.3. Here k_i was varied as per Eq. (36), so that the number of partial paths between interfaces $N_s^{(i)}$ was fixed to $\sim N_s = 18$ [i.e, $k_0 = 92$ (i=0) and $k_i = 5$ ($1 \le i < n$)] per run; these N_s and k_i' values were also chosen so that the cost C^{BG} matched that from the initial $\{\lambda\}$, unoptimized $\{k_i\text{'}\}$ simulations. Figure 2.6 showed approximately the same ν plateau values for BG simulations when the λ phase space was partitioned with n=9 and the optimized $\{k_i\text{'}\}$ or $\{\lambda_i\text{'}\}$ set was used. The contrast between Figure 2.6 and 2.8 illustrates the idea that the smaller the n is, the more inefficient an arbitrary choice of $\{\lambda_i\text{'s}\}$ is likely to be.

B. Genetic switch

This case illustrates the use of our approach for optimizing the $\{\lambda_i\}$ staging for a non-equilibrium, rare event process. This system has been used before to test FFS sampling methods[9,10,13] and it consists of a set of chemical reactions for a model symmetric bistable genetic switch whose dynamics do not obey detailed balance. The switch consists of a piece of DNA containing two genes ***A*** and ***B***, and an operator site ***O***. As described by the reaction set listed in Figure 2.9(a), the two genes encode for proteins ***A*** and ***B***, which can dimerize and bind to the operator site in their homodimer form. When $\boldsymbol{A_2}$ is bound to ***O***, ***A*** is produced at a rate k, but ***B*** is not produced. Likewise, when $\boldsymbol{B_2}$ is bound to ***O***, ***B*** is produced at a rate k, but ***A*** is not produced. A schematic illustration for the reactions and rate constants of this genetic switch is given in Figure 2.9(a).[9,10]

The FFS simulations were carried out using the Gillespie algorithm.[21] The order parameter was chosen to be $\lambda = N_B - N_A$, where N_A is the total number of ***A*** proteins, and N_B the total number of ***B*** proteins.[9,10] The probability distribution $P(\lambda)$ along this order parameter, shown in Figure 2.9, shows that there are two well defined

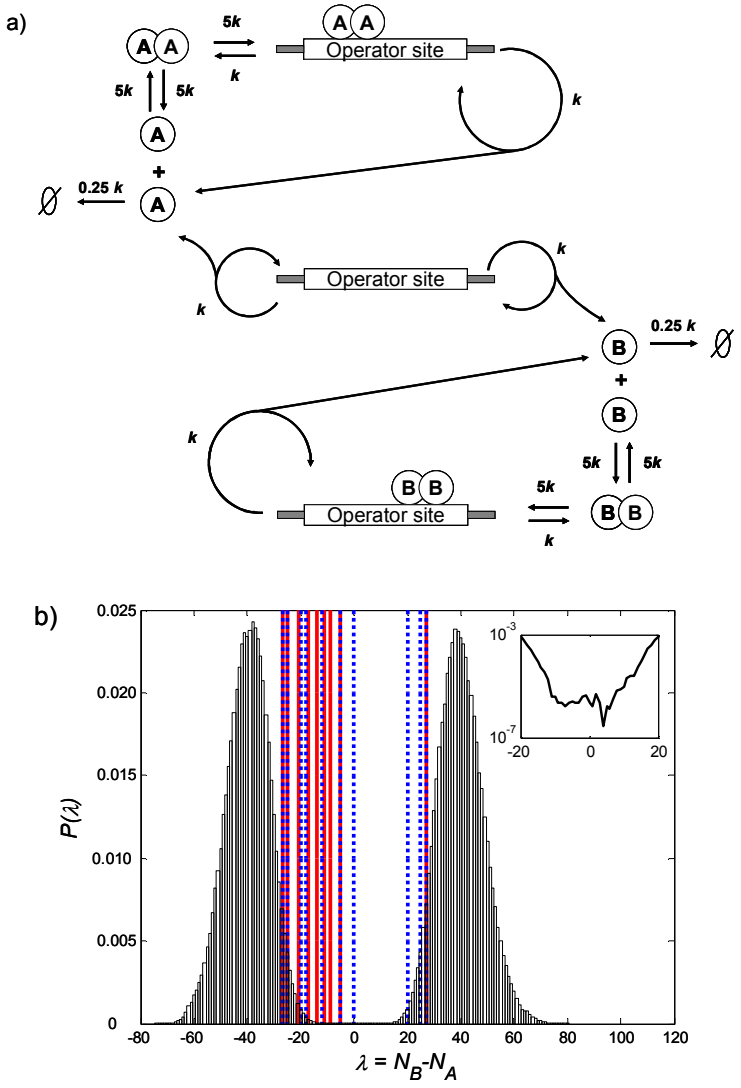

Figure 2.9 (a) Schematic illustration of the reactions involved in a model genetic switch that produces either protein A or B; a single operator molecule exists as either unbound, A_2-bound, or B_2-bound state. Forward and backward rate constants k_f and k_b are also given (Refs. 10 and 11). (b) Probability distribution $P(\lambda)$ for the genetic switch model (bars) from a straight brute-force simulation. The initial and final regions are labeled A and B, respectively. The initial λ staging for the FFS-type simulation is shown by dotted lines (blue). The thick line (red) shows the optimized $\{\lambda_i'\}$ staging. The inset shows, on a logarithmic scale, the profile of the region between the two basins (the lack of smoothness reflects the rarity with which the system visits this region).

steady states. Hence, the phase space region A was defined by taking $\lambda \leq -27$, and B by $\lambda \geq 27$, and the λ space was partitioned into n=8 interfaces. The calculations were carried out using the DFFS scheme as a series of blocks, each consisting of one FFS run. The number M of trials at each interface was fixed to $M_i = 5000$ ($0 \leq i < n$) and the starting points were randomly sampled from inside the region A. Results were averaged over ten blocks. For approach 2, the optimized $\{\lambda_i'\}$ set corresponds to fixing $P(\lambda_{i+1} \mid \lambda_i) = \left[P(\lambda_n \mid \lambda_0)\right]^{1/8} \approx 0.3$. Figure 2.9 and Table 2.4 show the initial and optimized $\{\lambda\}$ sets. Table 2.4 also shows the results for the values of $P(\lambda_{i+1} \mid \lambda_i)$ at each λ_i'. The optimization algorithm shifts the $\boldsymbol{\lambda}$ staging towards the "bottlenecks" of the FFS-type simulation, concentrating interfaces between the initial and TS regions where the vast majority of the trial runs per interface return to the initial basin of attraction A. We expected λ_{n-1} (= -5) to be located before the transition state $\lambda = N_B - N_A = 0$,[13] because $P(\lambda_{n=B} \mid \lambda_{n-1}) = \langle p_B \rangle_{\lambda_{n-1}} \approx 0.3$ at λ_{n-1}, which is smaller than the committor probability value at the TSE (i.e., $p_B \approx 0.5$). To see the effect on ν, simulations were run with the same number M of trials per point at λ_i (i.e., $M_i = 400$ for $0 \leq i < n$) for both the initial $\{\lambda_i\}$ set and the optimized $\{\lambda_i'\}$ (for the latter, the number of partial interfacial paths was $N_s^{(i)} = N_s = 100$ per starting point at λ_0, and $M_i = N_S / P(\lambda_{i+1} \mid \lambda_i) \approx 400$). Again, we observed a significant reduction in the statistical error [$\left(\nu_{un-optimized} - \nu_{optimized}\right) / \nu_{un-optimized} \approx$ (0.72-0.49)/0.72 $\approx$ 0.32] for the simulation with optimized staging.

For approach 2, we can verify that the optimized $\{\lambda_i'\}$ set is independent of the initial $\{\lambda_i\}$ set by performing a second optimization starting from the new set $\{\lambda_i\}$={-27, -24, -22, -20, -18, -15, -12, -9, -4, 27} ($0 \leq \lambda_i \leq n$=8). From these simulations (data not shown), we indeed find that the same $\boldsymbol{\lambda}$ distribution listed in Table 2.4 is obtained.

Table 2.4 Optimized $\{\lambda_i'\}$ set, and values of $P(\lambda_{i+1} \mid \lambda_i)$ for the genetic switch model. Results were averaged over ten blocks. The optimized set was found after three DFFS adaptive iterations.

	Initial $\{\lambda\}$ set	***Optimized* λ staging**	
i	λ_i	λ_i'	$P(\lambda_{i+1}\|\lambda_i)$
0	-27	-27	0.27
1	-25	-25	0.25
2	-18	-21	0.26
3	-12	-17	0.29
4	-5	-14	0.24
5	0	-11	0.27
6	20	-9	0.25
7	25	-5	0.27
8 (=n)	27	27	-

C. Lattice protein folding

Recently, we applied a FFS-type approach[13] denoted as FFS-LSE for estimating the reaction coordinate of a simple cubic lattice protein. The FFS-LSE algorithm uses the TPE of trajectories between the unfolded and folded states (generated by the BG scheme using initial order parameter) to obtain "on-the-fly" estimates of the committor probability to the final region, p_B, for all interfacial points. These p_B data are then used to screen a set of candidate collective properties for an optimized order parameter model λ (i.e., reaction coordinate) that depends on a few relevant variables. Least square estimation (LSE) is used to find the coefficients of the proposed model and an analysis of variance is used to determine the significant terms in the model.[13] We found that a simple approximation to p_B via a model linear on conformational energy (E) and number of native contacts (NC) was sufficient to describe the intrinsic folding dynamics of the proteins studied and to ensure an efficient sampling of the TPE. Now we explore the possibility that our adaptive optimization algorithm could aid the FFS-LSE optimization of the order parameter for this system (e.g., could a better λ staging for the initial order parameter more effectively generate the needed p_B history data?).

The 48-mer protein model adopted in this study has a unique, compact native structure whose energy is minimum $-E_{min}= 20.24\ k_BT$ and has $NNC = 57$ specific or "native" segment-segment contacts. Further details on the model protein, including its structure, thermodynamics and kinetic behavior are given elsewhere.[22-24] The folding dynamics of the system was simulated in free space at the corresponding *bulk transition temperature* $T_f = 0.27$.[23] The number of native contacts was used as initial guess of the order parameter (i.e., $\lambda=NC$) to partition the phase space between the unfolded and folded states:

Table 2.5 Optimized $\{\lambda_i'\}$ set, and values of $P(\lambda_{i+1} \mid \lambda_i)$ for the lattice protein model. Results were averaged over ten blocks. The optimized set was found after three DFFS adaptive iterations. The optimized set and values of $P(\lambda_{i+1} \mid \lambda_i)$ correspond to the third DFFS iteration. The $\langle p_B \rangle_\lambda$ values were approximated by Eq. (3).

	Initial {λ} set	***Optimized* λ staging**		
i	**λ_i**	**λ_i'**	**$P(\lambda_{i+1}\|\lambda_i)$**	$\langle p_B \rangle_\lambda$
0	9	9	0.16	~0.000
1	10	10	0.21	~0.000
2	12	12	0.24	0.001
3	15	15	0.18	0.004
4	20	17	0.22	0.021
5	26	20	0.32	0.096
6	29	24	0.72	0.301
7	33	30	0.85	0.418
8	37	32	0.90	0.492
9	41	34	0.91	0.547
10	46	37	0.77	0.600
11	51	44	0.78	0.780
12 (=n)	57	57		

$$\lambda_0 = \sqrt{NC^2 + E^2} = 9 \qquad (42)$$

$$\lambda_i = NC \qquad i > 0\,. \qquad (43)$$

Accordingly, we defined λ_0 as the upper limit of the unfolded state region (i.e., $\lambda_0 = \lambda_A$), defined by a circle centered at (NC=0, E=0) and radius of 9, while the final region B is the folded state $\lambda_B = \lambda_n = NNC$. We conducted a preliminary DFFS run to first optimize the $\{\lambda_i'\}$ set. We used initially n=12 interfaces to partition the phase space,[13] which were lineally distributed (i.e., $\Delta\lambda \approx$ constant) in λ_i ($0 \le i \le n$), as listed in Table 2.5 (and reported in Ref. 13). The initial $\{\lambda_i\}$ staging is also illustrated in Figure 2.10 (dotted lines). The number of trials per point at λ_i was M_i = 1000 ($0 \le i < n$) and results were averaged over ten blocks.

Table 2.5 shows the results for the probability values of $P(\lambda_{i+1} \mid \lambda_i)$ at each λ_i and the optimized $\{\lambda_i'\}$ set from the last (third) optimization iteration. Given that the FFS-LSE method will require a comprehensive p_B history, the optimized $\{\lambda_i'\}$ set was obtained by distributing the $P(\lambda_{i+1} \mid \lambda_i)$ values trying to target prescribed values of $\langle p_B \rangle_\lambda$ according to Eq. (3); i.e.,

$$P(\lambda_{i+1} \mid \lambda_i) = \langle p_B \rangle_{\lambda_i} \Big/ \langle p_B \rangle_{\lambda_{i+1}}, \qquad 5 \le i \le n-1 \qquad (44)$$

where we chose $\langle p_B \rangle_{\lambda_i}$ = {0.1, 0.3, 0.45, 0.5, 0.55, 0.6, 0.8} for $5 \le \lambda_i < n$-1. However, we chose $P(\lambda_{i+1} \mid \lambda_i) = P_\lambda$ for $0 \le \lambda_i \le 4$, where $P_\lambda = \left[P(\lambda_{n=B} \mid \lambda_0) / \prod_{i=1}^{n-1} P(\lambda_{i+1} \mid \lambda_i)\right]^{1/5}$ = constant; this guarantees that Eq. (2) is satisfied and that 5 interfaces are placed in the region close to the basin of attraction A (i.e., in steeply uphill region of $\langle p_B \rangle_\lambda \le 0.1$). Note that in this case, we used a completely different set of prescribed $P(\lambda_{i+1} \mid \lambda_i)$ values than the one used in the previous examples [i.e., Eq. (25) wherein $P(\lambda_{i+1}|\lambda_i)$ = constant for all i]. In this manner, we ensured that the λ interfaces are distributed over the entire phase space region: the region close to basin B ($10 \le \lambda_i \le$

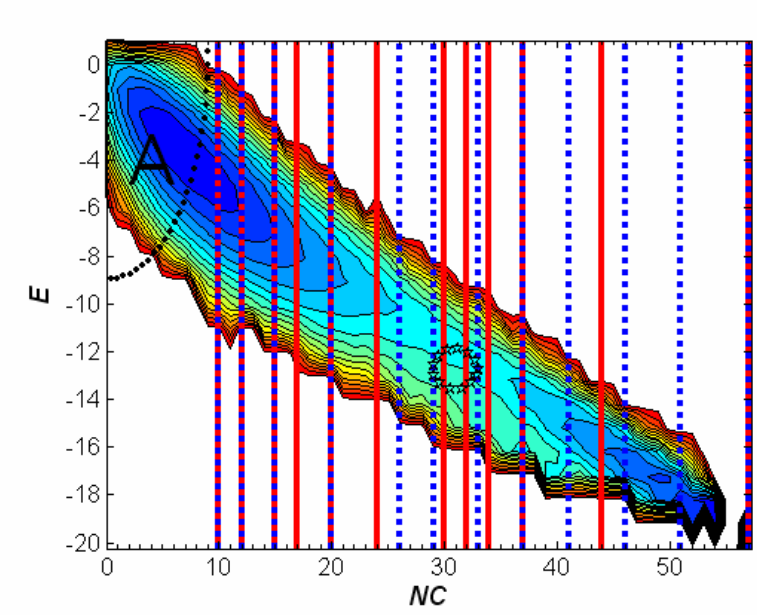

Figure 2.10 Staging in the DFFS simulations of folding of a lattice protein model. Contour plot of the free energy landscape [energy (E) vs. nativeness (NC)] is shown only for visual reference. The color scheme changes from highest (red) to lowest (blue) elevations. The initial region is enclosed by the circle labeled A. The initial λ staging is shown by dotted lines (blue) and the optimized staging by the thick lines (red). The most probable visited transition state region is enclosed by the ellipse with center at (μ_{NC}=31, μ_E =-12.73).[13]

12), the uphill region close to basin A ($0 \le \lambda_i \le 5$ for $\langle p_B \rangle_\lambda \le 0.1$), and the transition state (TS) region ($7 \le \lambda_i \le 9$ for $0.45 \le \langle p_B \rangle_\lambda \le 0.55$; in Figure 2.10 the TS is enclosed by the ellipse with center at $\{\mu_{NC}=31, \mu_E=-12.73\}$[13]). Note that the $\langle p_B \rangle_\lambda$ isosurfaces prescribed by Eq. (44) are only approximate but will approach the true p_B isosufaces if the order parameter used is a good reaction coordinate. The optimized $\{\lambda_i'\}$ set is also illustrated in Figure 2.10 (solid lines).

This optimized $\{\lambda_i'\}$ set was then used to obtain the p_B history data via BG simulations with the FFS-LSE method.[13] For this we also combined both optimization approaches; having already chosen a suitable λ staging, the optimized $\{k_i'\}$ set was found by letting $k_0'= 625$ and the other k_i' values to follow Eq. (31): $\{k_i\}=\{5, 5, 5, 5, 4, 3, 3, 3, 4, 5, 6\}$ for $1 \le i < n$. However, we set $k_i=5$ to be the minimum acceptable value needed to obtain p_B estimates with accuracy at the second significant digit (thus $k_i=5$ for $1 \le i \le n$-2). Following the FFS-LSE protocol, the p_B value for every interfacial point stored in the TPE trajectories was obtained by using recursively Eq. (5) from Ref. 13. Because p_B is the ideal reaction coordinate, a good order parameter model will be one that is able to "fit" well these p_B data. To find such a model, one assumes that p_B follows a mathematical relation (with linear coefficients) that depends on any number of candidate collective properties (suspected to be meaningful order parameters); the unknown coefficients are then found by standard least-square estimation and the statistically significant terms in the model are found by analysis of variance. The readers are referred to Ref. 13 for a detailed description of the FFS-LSE method. Coming back to the present case, the p_B data thus collected were fitted to a tentative regression model that included three collective variables: NC, E, and Rg^2 (the chain square radius of gyration), and interaction terms between these variables. As also confirmed in Ref. 13, the LSE and analysis of variance for this reaction

coordinate model indicated that the linear terms for *NC* and *E* are the only significant ones:

$$\lambda(NC, E) = -0.503 + 0.018(NC) - 0.032(E). \qquad (45)$$

Note that the model coefficients in Eq. (45) and the ones presented in Eq. (22) of Ref. 13 are not identical but similar (see Figure 2.11); the latter were obtained from a similar FFS-LSE approach but using n=12 equidistant interfaces and $k_i = 10$ for $0 \le i \le n\text{-}1$.[13] Equation (45), however, describes the variability of the p_B data with slightly higher statistical significance, having larger values for both the coefficient of multiple determination (R^2=0.87) and the P-value for the F statistic of test of lack of fit (0.985) compared to those of Eq. (22) from Ref. 13 (R^2=0.71 and P-value=0.965).[25] More importantly, the computational cost C^{BG} [estimated by Eq. (29)] required for the estimation of the reaction coordinate model in Eq. (45) is reduced by approximately two orders of magnitude (i.e., $C^{BG}[\text{Eq.(45)}]/C^{BG}[\text{Eq.(22) in Ref. 13}] \approx 10^{-2}$ and $\text{CPU}[\text{Eq.(45)}]/\text{CPU}[\text{Eq.(22) in Ref. 13}] \approx 10^{-2}$). This example illustrates that optimization of the λ staging can significantly reduce the computational effort of the FFS-LSE method in screening suitable reaction coordinate models. More generally, both λ-sampling optimization and order parameter optimization (via FFS-LSE) could be combined such that in each iteration the λ staging of the current order parameter is optimized and used to obtain a new estimate for the reaction coordinate, and this process repeated until a satisfactory convergence is attained (e.g., until the TS isosurface, p_B =0.5, coincides with the $\langle p_B \rangle_\lambda$ =0.5 interface from the stage optimization). For the example studied here, one iteration was enough to get suitable results.

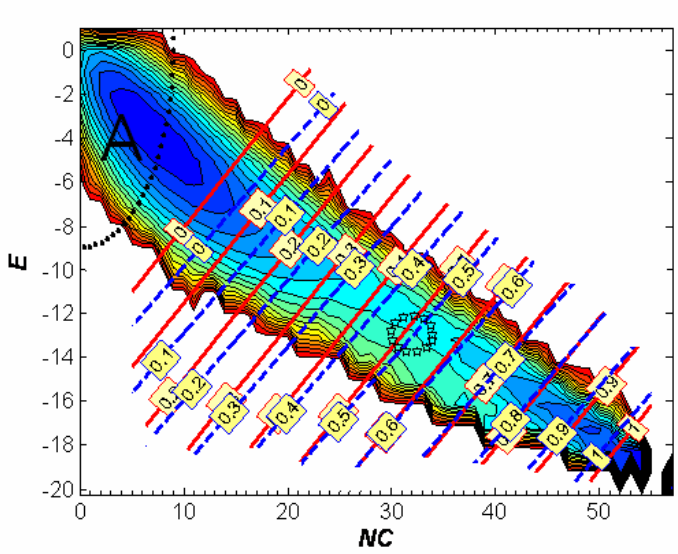

Figure 2.11 Estimated reaction coordinate iso-lines for the folding of the 48-mer lattice protein model at T_f=0.27. Thin lines (red) are the predicted p_B committors from Eq. (20) of Ref 13. Thick dotted lines (blue) are the predicted p_B committors from Eq. (45). The committor values are shown as labels. Contour plot of the free energy landscape [energy (E) vs. nativeness (NC)] is shown only for visual reference. The color scheme changes from highest (red) to lowest (blue) elevations. The initial region is enclosed by the circle labeled A. The most probable visited transition state region is enclosed by the ellipse.[13]

V. Conclusions

In this work, we have proposed a simple adaptive optimization algorithm for the numerical simulation of rare events in complex systems via FFS schemes. The optimization algorithm seeks to allocate the computational effort in a FFS simulation to reduce the statistical error with which the reaction rate constant is estimated from the connected paths between the stable states. By tracking the conditional probabilities to reach subsequent interfaces in the phase space we can identify the "bottlenecks" of the FFS-type simulation, typically occurring in the region between the initial and the transition state; feeding back this information we can concentrate the sampling on the bottleneck regions. Two different approaches were formulated: approach 1 that optimizes the number of fired trial runs per interface ($\{M_i'\}$or $\{k_i'\}$), and approach 2 that optimizes for the position (and number) of interfaces $\{\lambda'\}$.

We applied the proposed optimization approach to three simple test examples. The example of a particle moving on a bistable potential energy surface proved that even for this simple system our approach could optimize for both the $\{M_i'$ and $k_i'\}$ and the $\{\lambda_i'\}$ sets, allowing a significant reduction in the statistical error of the transition rate constant estimate. Furthermore, we showed that the method could also be used to determine the number of interfaces needed to partition the phase space in order to attain a desired crossing probability value between neighboring interfaces. The example on the genetic switch illustrated the application of the advocated methods to the simulation of nonequilibrium systems, rare-event problems. Finally, the lattice protein folding example showed that by optimizing the λ staging, the proposed methods can reduce the computational effort needed to obtain committor probability data for the estimation of refined reaction coordinate models via the FFS-LSE method.[13] This example also shows the convenience of the decoupling of both optimization algorithms (approaches 1 and 2): the λ phase staging was optimized first

to position interfaces at the TSE and FFS's "bottleneck" regions and the $\{k_i\}$ set was then optimized to improve the sampling in all important regions for a fixed computational cost.

While we used three iterations to ensure full convergence of the adaptive algorithm for all examples studied, one iteration was enough to get suitable convergence. Hence, for complex system where the FFS-type simulations are computationally costly, one iteration should be enough for optimize the sampling and staging. The formalism developed here to optimize the staging could be extended to any other interface-based path sampling methods [e.g., like milestoning methods,[5] transition interface sampling (TIS),[6,7]or partial path transition interface sampling (PPTIS)[8]].

Unlike FFS that targets kinetics data, expanded ensemble EXE methods are intended to extract thermodynamic functions (like free energies); however, both EXE and FFS rely on a "divide and conquer approach" to simulate more efficiently the passage between two end-states A and B by introducing intermediate states. Not surprisingly, similar strategies to those proposed here have been used for EXE to optimize either the state-sampling frequency or the location of those intermediate states (to concentrate the effort on transitional bottlenecks).[18,19] However, some key differences exist between FFS and EXE optimization methods. In EXE, non-Boltzmann biasing weights are used to control state-sampling frequency, while in FSS a varying number of starting trials is used to control sampling around any given interface (this way the natural dynamics of the system is preserved to get the TPE and rate constants). In EXE, it is often desirable to maximize the flow of round trips between end states, while in FFS it is rather a one-way flow process that needs to be optimized (e.g., by keeping a constant interfacial flow of trajectories). Finally, the type of order parameter used for EXE and FFS is typically different (e.g., for the former it

could be an unphysical parameter that need not depend on the system's coordinates); however, for some systems the optimized order-parameter found via FFS-LSE could also increase the efficiency or reduce hysteresis in EXE or multicanonical schemes.

While the simple examples studied here illustrate the validity and potential of the proposed method, we are currently pursuing applications to more complicated systems involving protein-protein interactions. For example, we are trying to determine the mechanism by which reassembly and folding occurs in split protein systems. Recently, we showed that the balanced distribution of the "folding nucleus"[13,23], (a subset of residues that are critical to the formation of the TS leading to productive folding), between protein fragments is key to their reassembly.[26] However, a more detailed study is needed to understand other characteristics of multichain protein assembly like the order of events. This type of analysis could be aided by such techniques as FFS-LSE that allows determination of a good reaction coordinate to describe the system's dynamics. However, we have found that its applicability is limited by sampling efficiency, which is dictated primarily by the initial choice of the order parameter but also by how the phase space is partitioned and sampled. Therefore, we hope that our adaptive optimization algorithm will become a valuable tool for studying the dynamics of complex systems where the efficiency of the path sampling approach limits the identification of a suitable reaction coordination.

Acknowledgments

The authors thank Francisco J. Martínez-Veracoechea for useful discussions. EB is grateful to the Sloan Foundation for a graduate Fellowship. Additional support from the National Science Foundation Award 0553719 is acknowledged.

REFERENCES

[1] P. G. Bolhuis, Proc. Nat. Acad. Sci. U.S.A. **100**, 12129 (2003).

[2] P. G. Bolhuis, D. Chandler, C. Dellago, and P. L. Geissler, Annu. Rev. Phys. Chem. **53**, 291 (2002).

[3] C. Dellago, P. G. Bolhuis, F. S. Csajka, and D. Chandler, J. Chem. Phys. **108**, 1964 (1998).

[4] C. Dellago, P. G. Bolhuis, and P. L. Geissler, Adv. Chem. Phys. **123**, 1 (2002).

[5] A. K. Faradjian and R. Elber, J. Chem. Phys. **120**, 10880 (2004).

[6] T. S. van Erp and P. G. Bolhuis, J. Comput. Chem. **205**, 157 (2005).

[7] T. S. van Erp, D. Moroni, and P. G. Bolhuis, J. Chem. Phys. **118**, 7762 (2003).

[8] D. Moroni, T. S. van Erp, and P. G. Bolhuis, Physica A **340**, 395 (2004).

[9] R. J. Allen, D. Frenkel, and P. R. ten Wolde, J. Chem. Phys. **124**, 024102 (2006).

[10] R. J. Allen, P. B. Warren, and P. R. ten Wolde, Phys. Rev. Lett. **94**, 018104 (2005).

[11] R. J. Allen, D. Frenkel, and P. R. ten Wolde, J. Chem. Phys. **124**, 194111 (2006).

[12] T. S. van Erp, J. Chem. Phys. **125** (2006).

[13] E. E. Borrero and F. A. Escobedo, J. Chem. Phys. **127** (16) (2007).

[14] B. A. Berg and T. Neuhaus, Phys. Rev. Lett. **68** (1), 9 (1992).

[15] P. Dayal, S. Trebst, S. Wessel, D. Wurtz, M. Troyer, S. Sabhapandit, and S. N. Coppersmith, Phys. Rev. Lett. **92** (9) (2004).

[16] S. Trebst, D. A. Huse, and M. Troyer, Phys. Rev. E **70** (4) (2004).

[17] S. Trebst, M. Troyer, and U. H. E. Hansmann, J. Chem. Phys. **124** (17) (2006).

[18] F. A. Escobedo, J. Chem. Phys. **127** (17) (2007).

[19] F. A. Escobedo and F. J. Martinez-Veracoechea, J. Chem. Phys. **127** (17) (2007).

[20] B. Peters and B. L. Trout, J. Chem. Phys. **125** (2006).

[21] D. Frenkel and B. Smit, *Understanding Molecular Simulation: From Algorithms to Applications*, 2nd ed. (Academic, Boston, 2002).

[22] V. I. Abkevich, A. M. Gutin, and E. I. Shakhnovich, Fold Des. **13**, 221 (1996).

[23] E. E. Borrero and F. A. Escobedo, J. Chem. Phys. **125**, 164904 (2006).

[24] L. M. Contreras, F. Martinez-Veracoechea, P. Pohkarel, A. D. Stroock, F. Escobedo, and M. P. DeLisa, Biotechnol. Bioeng. **94**, 105 (2006).

[25] D. G. Mongomery, *Desing and Analysis of Experiments*, 5th ed. (John Wiley & sons, Inc., New York, 2001).

[26] L. M. Contreras, E. E. Borrero, F. A. Escobedo, and M. P. DeLisa, Biophys. J . **94**, 1 (2008).

3. CHAPTER: Simulating the kinetics and thermodynamics of transitions via a forward flux/umbrella sampling*

Abstract

Firstly, a technique is introduced for computing equilibrium probability distributions for transitional rare-event simulations by combining the ensemble of trajectories generated by forward flux sampling (FFS) and by umbrella sampling (US) in multiple windows along an order parameter of interest; this method is denoted FFS-US. Secondly, the FFS algorithm is extended to obtain rate constants of partial transitions involving intermediate states from a single simulation; this is denoted "multiple state" FFS. For the FFS-US method, a FFS algorithm (pre-optimized for order parameter and staging) is used to take advantage of its zero-potential-bias of phase-space sampling to gather histogram data with which jump-start the US and get the equilibrium distributions. In this way, kinetic data (like the rate constants and the transition path ensemble) and the underlying free-energy landscape (or probability distribution) of the system are obtained efficiently and concurrently. The applicability of these techniques is illustrated by studying several test systems, including two that involve potential energy surfaces having multiple metastable states and transition pathways, representative of complex kinetic behavior.

I. Introduction

Many processes in complex molecular systems involve rare events; e.g., in phase transitions and conformational changes between stable or metastable states. Simulating the transition kinetics (i.e., rate constants and mechanistic details) and the relative free energies of such metastable basins and those associated with intervening

---

* E. E. Borrero and F. A. Escobedo, J. Phys. Chem. B (submitted).

barriers are thus central objectives in many applications of computational physical-chemistry. However, conventional numerical techniques are impractical for the simultaneous computation of free energies and kinetics of rare events. Schemes such as forward flux sampling[1, 2] (FFS) allow the computation of rate constants by overcoming the problem associated with simulating rare events (i.e., enhancing the sampling by a series of interfaces that partition the phase space and reducing the CPU time wasted on the uneventful waiting time between events), while techniques such as umbrella sampling[3] (US) allow the calculation of free energy barriers separating the stable states. In this article, we explored the idea of using the framework of FFS to evaluate both the kinetics of the transition pathways and the underlying free energy landscape. The proposed approach to be denoted FFS-US, computes free energies using the ensemble of trajectories generated by a FFS scheme combined with US data for windows defined by the FFS interfaces. Essentially, FFS-US combines the zero-potential-bias data from FFS with additional US data to calculate the potential of mean-force along a chosen reaction coordinate (i.e., order parameter).

Other research groups have proposed schemes for the simultaneous computation of free energies and kinetics of rare events. For example, Radhakrishnan and Schlick[4] developed "BOLAS", a scheme for free energy calculations, by combining the Monte Carlo (MC) ensemble of trajectories from the shooting algorithm of transition path sampling (TPS) with the window-based US strategy to enhance the efficiency of computing the probability density distribution $p(\lambda)$ (i.e., the probability to find the system at a certain value of the order parameter λ) over a desired range of the reaction coordinate. Moroni et al.[5] also introduced a method to evaluate simultaneously the transition rate constant and the free energy profile using partial path transition interface sampling (PPTIS) scheme. Their method corrects the bias introduced in the path ensemble obtained by the PPTIS formalism by comparing

neighboring interface ensembles. For this purpose, suitable weights are calculated for the histograms of points visited around the outer interfaces for two consecutive windows and used to scale the ensembles of states visited around the inner interface. The resulting histograms of the probability density p (λ) on each window are matched together to obtain the continuous $p(\lambda)$ distribution. The milestoning method proposed by Faradjian and Elber [6] also employs a series of interfaces to estimate the average transition rate constant and $p(\lambda)$, assuming that the partial transitions between consecutive interfaces do not depend upon the full history of the path. Both PPTIS and milestoning are applicable to systems in thermodynamic equilibrium. Ytreberg and Zuckerman [7] developed the "black-box" strategy for reweighting any ensemble of arbitrarily generated configurations to produce a canonically distributed ensemble and estimate free energy differences.

Recently, Valeriani et al.[8] introduced a method for computing stationary distributions (i.e., the "free energy" profile and the steady-state probability distribution for equilibrium and nonequilibrium systems, respectively) using FFS schemes. In their method, the stationary distribution $P_{ss}(\lambda)=P(\lambda)$ along a reaction coordinate λ is obtained by performing two FFS simulations to obtain the rate constants for the forward and backward transitions. These rates are then used to reweigh contributions to $P(\lambda)$ from trajectories originating from both region A and region B. The need of performing these two FFS simulations can be a shortcoming of this method because in many applications only one transition is of interest (e.g. the folding of a protein as opposed to unfolding) and if one state is much more stable than the other, then sampling the transition toward the less stable state may be computationally very demanding. A related method to estimate $P(\lambda)$ has also been proposed [9] by matching histograms from backward and forward reactions in transition interface sampling (TIS) simulations.

The proposed FFS-US method can be seen as complementary to that of Valeriani et al.[8] wherein rather than performing an additional FFS simulation (for the reverse transition), the US strategy is implemented following an original FFS protocol. Once the FFS run has ended, the transition path ensemble is reweighted (following a scheme akin to that of Moroni et al.[5]) and the US takes over to sample the regions inside the corresponding windows w_i (bounded by the hard walls at interfaces λ_i and λ_{i+1}) until the partial path ensemble loses any "memory" of where it originated. However, the FFS-US method is more limited than Valeriani et al.'s method in that the former is only indicated for equilibrium probability distributions. Wamflash et al.[10] has presented an algorithm that can also determine the steady-state probability distribution for nonequilibrium processes by transferring information about fluxes and probabilities between neighboring US regions (to correct for the lack of detailed balance). While a FFS approach could potentially be designed to extract such information (and thus generalize the uses of FFS-US), this lies beyond the scope of this work.

A second goal of this paper is to extend the original FFS formalism to study transitions that involve intermediate state by using concepts similar to those of the multiple state transition path sampling method by Rogal and Bolhuis.[13] In the multiple state FFS method, the rate constant estimates for all forward reactions including transitions that go from basin A to B through intermediates states and from A to any intermediate state, are calculated from one single simulation rather than from separate FFS simulations for each partial reaction.

This paper complements two previous articles in which we addressed several challenges of applying a FFS-type algorithm for the simulation of complex system; namely: (i) a method to determine an adequate reaction coordinate (i.e., order parameter) to describe the system's dynamics (denoted FFS-LSE)[11] and (ii) an

adaptive algorithm to optimize for either the number and position of the interfaces (i.e., optimized λ phase staging), and/or the number of fired trial runs per interface[12] to reduce the statistical error in the rate constant estimation (for a given computational cost). Our strategy here is thus to take full advantage of this platform - the optimized order parameter, staging, and collection of transition paths obtained from a FFS simulation - to set up a complementary US run to obtain free-energy information.

By way of background, we start by briefly reviewing the FFS-type simulation scheme for the calculation of rate constants and transition pathways (Sec. II A) and the US protocol (Sec. II B). In Secs. II C and D, we discuss how the US was implemented within the FFS framework to map the energy landscape. In Sec. II E, we introduce the formalism for multiple state FFS simulations. In Sec. III, we apply the FFS-US approach to several test cases including two-dimensional rugged energy surfaces that are representative of numerous complex systems. We also compare in this section the thermodynamic results for the test systems with those from straight brute-force calculations. In Sec. IV, we provide some concluding remarks.

II. Methods

A. Forward Flux Sampling (FFS)

In this work, we focus the discussion on two of the three path sampling schemes proposed in the literature to generate transition paths by a FFS-type approach: (i) the Direct Forward Flux Sampling (DFFS), and (ii) the Branched Growth method (BG).[1, 2] Essentially, these schemes sample the transition path ensemble (TPE) by performing MC sampling of dynamic trajectories between interfaces. The rate constant of the process is estimated as an average rate of transitions from two well-defined states A and B using an "effective positive flux" expression.[1, 2, 14, 15] An order parameter $\lambda(x)$ (where x is the phase space coordinates) is used to partition the phase

space by employing a series of nonintersecting interfaces (n+1) such that the system is considered in region A for $\lambda(x) \leq \lambda_A(x)$ and in region B for $\lambda(x) \geq \lambda_B(x)$. The effective flux expression estimates the rate constant $k_{A \to B}$ for transitions from A to B as:[1, 2]

$$k_{A \to B} = \overline{\Phi}_{A,0} P(\lambda_{n=B} \mid \lambda_0) \qquad (1)$$

where $\overline{\Phi}_{A,0}$ is the total average flux of trajectories from A to λ_0, and $P(\lambda_{n=B} \mid \lambda_0)$ is the probability that a trajectory reaching λ_0 from A will reach to B without returning to A.[1] Likewise, $P(\lambda_{n=B} \mid \lambda_0)$ can be expressed as the product of conditional probabilities:

$$P(\lambda_{n=B} \mid \lambda_0) = \prod_{i=0}^{n-1} P(\lambda_{i+1} \mid \lambda_i) \qquad (2)$$

where $P(\lambda_{i+1} \mid \lambda_i)$ is the probability that a trajectory that visits A and crosses λ_i for the first time will subsequently reach λ_{i+1} without returning to the initial region A.[12] For a complete description of the theoretical background of the algorithm, see Ref. 1.

B. Umbrella sampling (US)

The potential of mean force $F_i(\lambda)$ is calculated by measuring the probability distribution $p_i(\lambda)$ to be at a certain value of the order parameter λ chosen a priori to describe the transition between basins A and B:

$$F_i(\lambda) = -k_B T \ln[p_i(\lambda)] + const. \qquad (3)$$

To enhance the efficiency of computing $p_i(\lambda)$ over the desired range of λ, we essentially performed a window-based US strategy. The desired range in the phase space is divided up in smaller windows, w_i ($\lambda_0 \leq \lambda_i \leq \lambda_{n-1}$), and the walls are used as hard boundaries [i.e., states are accepted only if they visit the region inside the window]. This is equivalent to performing an US with a weighting function of zero

inside the window. The functions $F_i(\lambda)$ in different windows are pieced together by matching the constants such that the entire function $F(\lambda)$ is continuous at the boundaries of the windows. Hence, after a predetermined number of statistics per window are accumulated, the un-normalized $p(\lambda)$ distribution can be estimated recursively[16]

$$\frac{p(\lambda)}{p(0)} = \frac{H_{0r}}{H_{0l}} \cdot \frac{H_{1r}}{H_{1l}} \cdots \frac{H_i(\lambda)}{H_{il}} = \prod_{n=1}^{i-1} r_i \cdot \frac{H_i(\lambda)}{H_{il}} \quad , \qquad (4)$$

where $\lambda \in [\lambda_i , \lambda_{i+1}]$. A histogram $H_i(\lambda)$ monitors how often each state is visited in the ith window $[\lambda_i , \lambda_{i+1}]$, where λ_i is the interface in which trial runs are initiated in the FFS method. The log of ratios $r_i = \frac{H_{ir}}{H_{il}}$ in Eq. (4) correspond to free energy differences, and $H_{il} \equiv H_i(\lambda_i)$ and $H_{ir} \equiv H_i(\lambda_{i+1})$ denote the values of the ith histogram at its left and right boundary, respectively. When evaluating $H_i(\lambda)$, if a move attempts to leave the w_i, it will be rejected and H [at the window edge] incremented by one to fulfill detailed balance.[8, 16] However, if the location of a window (in the free-energy space) is such that a strong force drives the system toward the front or back wall, the method could lead to artificial spikes in probability density at a window edge. To avoid this, one could increment the nominal window's size a little to $[\lambda_i - \Delta\lambda , \lambda_{i+1} + \Delta\lambda]$, and only count statistics inside the $[\lambda_i , \lambda_{i+1}]$ region for the $H_i(\lambda)$ histogram.

C. N_{bin} and $N_{win}^{(i)}$

The overall error in $p_i(\lambda)$ depends on the choice of the number of statistics accumulated per window i ($N_{win}^{(i)}$) and the minimum number of entries per bin (N_{bin}). $N_{win}^{(i)}$ is essentially the time t_{w_i} (in units in number of steps) needed to obtain a predetermined relative statistical error in $p_i(\lambda)$, estimated to be:

$$N_{win}^{(i)} \approx t_{w_i} = a\omega_i^2 \quad , \quad (5)$$

where ω_i is the window size and a is a proportionality constant.[16] If all the windows are the same size $\omega_i=\omega$, the total computational time to sample the complete phase space is:

$$t_{CPU} = \sum_{i=1}^{m} t_{\omega_i} = a\sum_{i=1}^{m} \omega_i^2 = am\omega^2 \quad , \qquad (6)$$

where m denotes the number of windows into which the phase space was subdivided. In the limit of a single large window (i.e., m=1 and ω'=$m\omega$), the total computational time for a standard brute force (BF) simulation is:

$$t'_{CPU} = a\omega'^2 = a(m\omega)^2 = mt_{CPU} . \quad (7)$$

Showing the known fact that the basic window strategy reduces the computational effort by a factor of m.[3, 16]. Hence, for a fixed number of windows and an estimated t'_{CPU} value, $N_{win}^{(i)}$ is calculated from Eq. (5), where the constant a is estimated using Eq. (7) and (6). Sampling inside the window's region, w_i, should therefore be continued until satisfying the condition:

$$\sum_{\lambda=\lambda_i}^{\lambda_{i+1}} H_i(\lambda) \geq N_{win}^{(i)} . \quad (8)$$

While $N_{win}^{(i)}$ sets a bound on the number of statistics needed for each histogram in window w_i , N_{bin} sets a limit for the minimum amount of local sampling (in a given

bin). To estimate N_{bin}, we assume that the number of points per bin follows a Poisson distribution and therefore the relative statistical error (δ) in $p_i(\lambda)$ is given by[17]

$$\delta \propto \left[N_{bin}\right]^{-1/2} \qquad (9)$$

from which it is clear that δ always decreases as additional statistics are added to the histogram $H_i(\lambda)$ in the w_i. Thus, for any one l bin, N_{bin} should be in the range [10^2, 10^4] in order to obtain a δ of order $\Delta_l \sim O(10^{-2})$. Based on this, we assumed that at each interface "memory effects" may persist for longer than the time needed to travel from one interface to the other (i.e, from λ_i to λ_{i-1} or λ_{i+1}), but not much longer than the time required to satisfy the condition:

$$\min[H_i(\lambda)] \geq N_{bin}, \qquad (10)$$

i.e., this condition implies that the overall error Δ for $H_i(\lambda)$ is of order $\Delta \sim O(\Delta_l)$.
In summary, the correct probability distribution in the i^{th} window is obtained by accumulating statistics in $H_i(\lambda)$ until the conditions in Eq. (8) and Eq. (10) are met. In Sec. III, we discuss how the values for $N_{win}^{(i)}$ and N_{bin} correlate to the statistical error in $p_i(\lambda)$.

D. FFS-US combination

In this section, we introduce the proposed method, FFS-US, for efficient calculation of probability distributions following the simulation of the rate transition constant and the transition state ensemble (TPE). The FFS-type formalism generates partial paths between interfaces by initiating trial trajectories from an interface λ_i, and ending when the next interface λ_{i+1} or the basin A (i.e., λ_0) is hit. Even though the harvested partial paths started at λ_i are free to follow any possible path between A and

λ_{i+1}, including paths crossing previous interfaces several times, this sampling is incomplete because the partial paths reaching λ_{i+1} are stopped and thus not allowed to explore the system in the opposite direction (i.e., unidirectional sampling). Even statistics from partial paths returning to the initial region A introduce a bias, since the $p_i(\lambda)$ distribution would only contain contributions from trajectories coming from the basin A, missing the contribution from those that originate from basin B. To correct for these biases, a complementary US strategy is performed by accumulating more statistics between windows. While this US portion could be done concurrently at each stage of the FFS simulation, for simplicity, we assume here that US run is performed *after* the FFS run has finished. For this purpose, the FFS contribution to $H_i(\lambda)$, to be denoted as the histogram $h_i^{FFS}(\lambda)$ must be stored during the FFS run as explained shortly. Hence, the total histogram $H_i(\lambda)$ is computed by joining $h_i^{FFS}(\lambda)$ and an extra US histogram $h_i^{US}(\lambda)$:

$$H_i(\lambda) = h_i^{FFS}(\lambda) + h_i^{US}(\lambda) \quad (11)$$

D.1 Evaluation of $h^{FFS}(\lambda)$. During the FFS-type simulation, for each trial trajectory initiated at λ_i, statistics of the system's visits inside the boundaries ($\lambda_{i-1} < \lambda_i < \lambda_{i+1}$) are accumulated until the system reached the λ_{i-1} or λ_{i+1} interface. The collection of all states "visited" by these paths is a subset of the phase space points between λ_{i-1} and λ_{i+1} because the points from trajectories meandering around the outer interfaces are missing. In analogy to the method presented by Moroni et al.,[5] the states along a partial path are categorized as either "loop" or "boundary" points as illustrated in Figure 3.1. For any given trajectory started at λ_i, the states connecting λ_i and λ_{i-1} are classified as backward boundary points $\left[b_{i\to i-1}(\lambda)\right]$ and the rest of them as loop points $\left[l_i(\lambda)\right]$. Likewise, for any given trajectory started at λ_i, the states connecting λ_i

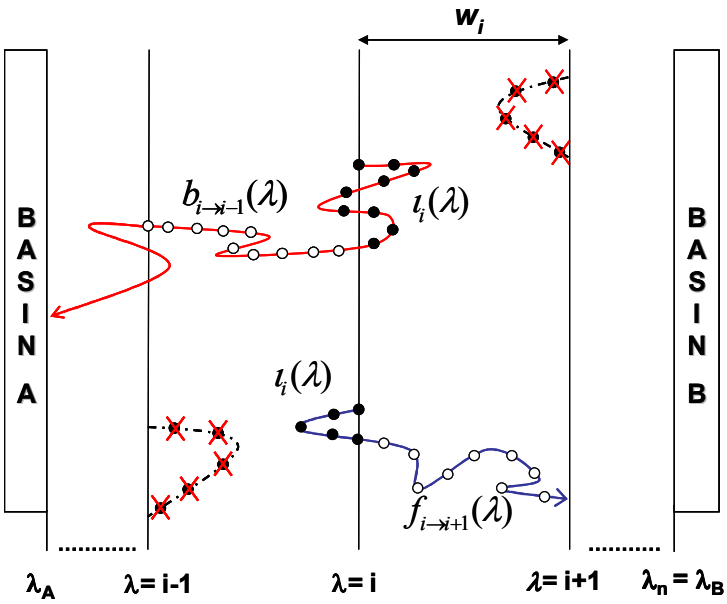

Figure 3.1 A schematic view of loop and boundary points during a FFS run. Boundary points (open circles) connect a boundary interface (λ_{i-1} or λ_{i+1}) with the middle interface (λ_i). The loop points (filled circles) belong to trajectories that meet first the middle interface in both directions. The dashed lines and crosses correspond to paths that are not sampled during FFS.

and λ_{i+1} are classified as forward boundary points $\left[f_{i\to i+1}(\lambda)\right]$ and the rest of them as loop points $\left[l_i(\lambda)\right]$. Any lack of sampling around the outer interfaces can then be made up for by meshing neighboring interface ensembles as follows. First, note that in steady-sate the flux of trajectories coming from A and B should be the same; i.e.:

$$n_{i\to i+1} = n_{i+1\to i} \ , \qquad (12)$$

where $n_{i\to i+1}$ is the number of partial paths started at λ_i and meet λ_{i+1} before λ_{i-1}, and $n_{i+1\to i}$ is the number of partial paths started at λ_{i+1} and meet λ_i before λ_{i+2}. Hence, the FFS run contribution to the total $H_i(\lambda)$ histogram can be estimated by weighing the contributions of neighboring interfaces ensembles to satisfy Eq. (12); i.e., for points in $[\lambda_i, \lambda_{i+1}]$ we have:

$$h_i^{FFS}(\lambda) = f_{i\to i+1}(\lambda) + W_{i-1}\left[l_i(\lambda)\right] + W_i\left[b_{i+1\to i}(\lambda) + l_{i+1}(\lambda)\right] \ , \quad (13)$$

where the scaling factors are defined as $W_0 = 1$ and $W_i = n_{i+1\to i} / n_{i\to i+1}$. The W_{i-1} term in Eq. (13) arises from the matching procedure for the loop and boundary point histograms for the previous $\lambda_{i-1} \le \lambda \le \lambda_i$ window. Once all scaling factors (W_i) are known, the total histogram $h_i^{FFS}(\lambda)$ can be computed by joining all the boundary and loop histograms via Eq. (13).

D.2 Evaluation of $h^{US}(\lambda)$. The US contribution to the total $H_i(\lambda)$ histogram in the w_i window ($\lambda_i \le \lambda \le \lambda_{i+1}$) is obtained from a conventional US simulation using the interfaces as hard window boundaries (λ_i and λ_{i+1}) with a weighting function of zero.[4] For each window, this simulation is initiated from N_{US} points at interface λ_i, randomly selected from those stored during the FFS run, and continued until Eqs. (14) and (15) below [which are essentially Eqs. (8) and (10)] are satisfied. Equation (14) is Eq. (8) suitably modified in account of Eq. (11):

$$\sum_{\lambda=\lambda_i}^{\lambda_{i+1}} h_i^{US}(\lambda) \geq N_{win}^{(i)} - \sum_{\lambda=\lambda_i}^{\lambda_{i+1}} h_i^{FFS}(\lambda) \qquad (14)$$

where $h_i^{US}(\lambda) = \sum_{k=1}^{N_{US}} h_{i,k}^{US}(\lambda)$ and $h_{i,k}^{US}(\lambda)$ is the US histogram registering entries in the i^{th} window and k^{th} starting point at λ_i. Equation (15) is Eq. (10) applied to the US data only:

$$\min[h_{i,k}^{US}(\lambda)] \geq N_{bin}, \qquad \forall k, \qquad (15)$$

For simplicity, we used N_{US}=1 for the examples in this work. As discussed in Sec. II C, N_{bin} is chosen such that the paths sample w_i long enough to lose "memory" of their starting points, while $N_{win}^{(i)}$ is chosen such that the total computational cost to obtain the equilibrium distribution from the FFS-US approach is comparable to that of a conventional US simulation.

The unnormalized distribution inside each window (w_i) is simply found from $p_i(\lambda) = H_i(\lambda) / \sum_{\lambda=\lambda_i}^{\lambda=\lambda_{i+1}} H_i(\lambda)$. To obtain the complete $p(\lambda)$ distribution, we need to employ conventional US calculations to sample the states inside and close to the stable regions A and B where $p(\lambda)$ can be readily obtained this way. The FFS-US is therefore reserved to sample $p(\lambda)$ in the barrier region between A and B which is rarely visited by brute-force "BF" simulations (i.e., leading to poor statistics). In practice, performing FFS-US only on a few windows with some overlapping (i.e., if neighboring windows overlap by at least one state) is sufficient to obtain an entire continuous free energy profile in the region [λ_0, λ_{n-1}], as shown in Sec III.

The contribution to the w_i's $H_i(\lambda)$ from the $h_i^{FFS}(\lambda)$ is defined as the ratio:

$$\mathrm{R_{FFS}} = \sum_{\lambda=\lambda_{\mathrm{i}}}^{\lambda_{\mathrm{i+1}}} h_{\mathrm{i}}^{\mathrm{FFS}}(\lambda) \Big/ \sum_{\lambda=\lambda_{\mathrm{i}}}^{\lambda_{\mathrm{i+1}}} H_{\mathrm{i}}(\lambda) \qquad (16)$$

where the $h_i^{FFS}(\lambda)$'s only include statistics from FFS trajectories that contributed to the $p(\lambda)$ distribution. R_{FFS} can be seen as a measure of the efficiency of the FFS-US combination since it represents the fractional savings in the length of the standard US simulation that the FFS data provides (the higher the R_{FFS}, the higher the efficiency).

It is finally noted that rather than using histogram-based or "visited-states" approach to find the $p(\lambda)$ distribution [i.e., accumulating histograms for statistics of the frequency with which the system visits λ states] we could employ a transition matrix (TM) method that relies on information on the probabilities of transitioning between λ states.[18] While we did implement such a TM approach (following Ref. [18]) and found its results to agree well with those of visited-states approach, we didn't observe any clear advantage of using former and thus we'll only report results for the latter.

E. Multiple state FFS

In FFS simulations, the rate constant estimation is restricted to pathways connecting two stable states. However, for complex systems the trajectories between two stable states might visit several intermediate states interconnected in phase space. If we are interested in estimating rate constants for all the multiple transitions, $N(N\text{-}1)$ (where N is the number of metastable states) independent FFS simulations could be performed, but such an approach would be very inefficient. To partially address this problem, we reformulated the FFS formalism to calculate rate constant for all the transitions that connect any two stable states through intermediate states within one single simulation. In analogy to the method presented by Rogal and Bolhuis,[13] it is assumed that the intermediate states have already been identified and that we can express the rate constant for transitions from state A to B and passing through M metastable states forming a set **m** (i.e., $M \geq 1$) as

$$k_{A\to\mathbf{m}\to B} = P(\lambda_{n=B} \mid \lambda_{\mathbf{m}})\, k^{T}_{A\to B} \qquad (17)$$

where $k^{T}_{A\to B}$ is the overall transition rate constant connecting states A and B, and $P(\lambda_{n=B} \mid \lambda_{\mathbf{m}})$ is the conditional probability that whenever the interfaces of the intermediate states **m** are crossed by a trajectory coming from A, state B will be reached before returning to A (note that **m** could denote a single or multiple metastable states). The intermediate state crossing probability $P(\lambda_{n=B} \mid \lambda_{\mathbf{m}}) = n_{A-\mathbf{m}} / N^{\{n\}}$ is simply given by the number, $n_{A\text{-}\mathbf{m}}$, of pathways starting in A, crossing the intermediate states **m** and ending in B divided by the number of all the pathways, $N^{(n)}$, starting in A and reaching B. Equation (17) is further simplified using Eq. (1) and recalling that for the BG method $P(\lambda_{n=B} \mid \lambda_0) = N^{\{n-1\}} / \prod_{i=0}^{n-1} k_i$:

$$k_{A\to m\to B} = \Phi_{A0} \frac{n_{A-\mathbf{m}}}{\prod_{i=0}^{n-1} k_i} \qquad (18).$$

Once the FFS simulation ends, $n_{A\text{-}\mathbf{m}}$ is obtained by tracing back the pathways in the TPE. Since the flux, $\overline{\Phi}_{A,0}$, is constant for all transitions out of state A, the ratio of two rate constant can be expressed as the ratio of intermediate state crossing probabilities,

$$\frac{k_{A\to\mathbf{m}\to B}}{k_{A\to\mathbf{r}\to B}} = \frac{P(\lambda_{n=B} \mid \lambda_{\mathbf{m}})}{P(\lambda_{n=B} \mid \lambda_{\mathbf{r}})} = \frac{n_{A-\mathbf{m}}}{n_{A-\mathbf{r}}}. \qquad (19)$$

There are two types of rate constants of particular interest to understand the transition mechanism: (i) Rate constants $k_{A\to m}$ where m is a single intermediate state visited directly from A; these tell us about the possible destinations of pathways right after leaving A, and (ii) rate constants $k_{A\to *\to m\to B}$, which are instrumental to quantify the immediate sources of trajectories that reach B. Here the star symbol (∗) is used to

represent any number of possible jumps among different intermediate states, so that A$\rightarrow$*$\rightarrow m\rightarrow$B denotes the transition for all trajectories that started in A, ended in B, and visited intermediate state "m" just before ending in B, regardless of whether or not jumps among intermediate states occurred before the $m\rightarrow$B step. Note that $k_{A\rightarrow B}+\sum_{m=1}^{M}k_{A\rightarrow m}$ is the total rate of leaving A to any other destination, and the total A$\rightarrow$B rate constant is $k_{A\rightarrow B}^{T}=k_{A\rightarrow B}+\sum_{m=1}^{M}k_{A\rightarrow *\rightarrow m\rightarrow B}$, where $k_{A\rightarrow B}$ is the rate constant associated with direct paths from A to B (without passing through any intermediate).

To evaluate such rate constants as $k_{A\rightarrow m}$ and $k_{A\rightarrow *\rightarrow m\rightarrow B}$, it is convenient to label the stored points as $\{p,q\}$, keeping track of the two most recent states visited: "q" is the most recent state visited and "p" is the one visited before q. The $\{p,q\}$ pair is initialized as $\{a,a\}$ where "a" denotes basin A as the origin. State "p" could be either another intermediate state ($p\neq q$) or state A ($p=a$, when the path went A$\rightarrow q$ directly). State "q" could either be an intermediate state or still be $q=a$ if no intermediate state has yet been visited. Thus, e.g., if a partial trajectory connects a configuration at λ_i with pair label $\{p_i,q_i\}$ with a configuration at λ_{i+1} with label $\{p_{i+1},q_{i+1}\}$, then if such partial trajectory visited intermediate state m, we must have $p_{i+1}=q_i$ and $q_{i+1}=m$; otherwise the label stays the same (i.e., $p_{i+1}=p_i$, $q_{i+1}=q_i$).

To evaluate constants $k_{A\rightarrow m}$, rather than using separate FFS runs, we can estimate them via a single FFS simulation (connecting stable states A and B) as follows. The stored states at λ_i are categorized as either "active" (a) or "inactive" ($\cancel{a}$) points as illustrated in Figure 3.2(b). An active point in one for which the $\{p,q\}$ label is $\{a,a\}$, and is inactive otherwise. Clearly, an "active" point is a state along a partial trajectory that starts in basin A and has not committed yet to any intermediate or stable state. Figure 3.2(a) shows schematically that the overall transition rate constant $k_{A\text{-}m}$ can then be estimated as the treatment for reactions or resistors in parallel:

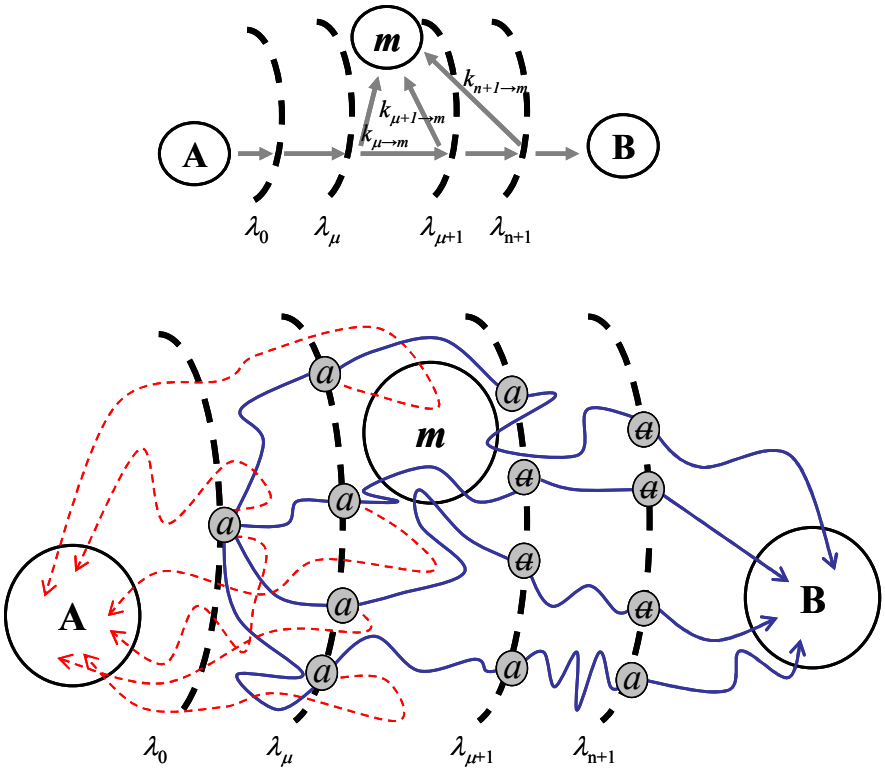

Figure 3.2 Multiple state FFS simulations. (a) The overall transition rate constant between the basin A and m intermediate state, kA-m, is estimated as resistors in parallel: $k_{A\to m} = k_{\mu\to m} + k_{\mu+1\to m} + k_{n+1\to m}$. (b) A schematic view of the "active" (a) and "inactive" (α) points for the calculation of the rate constants for multiple transitions.

$$k_{A\to m} \approx \sum_{\xi=\mu}^{n-1} k_{\xi\to m}\ , \quad (20)$$

where ξ corresponds to the interface index and μ is the index of last interface in the phase space immediately prior to the intermediate state *m*. To calculate the rate constant between a ξ interface and intermediate *m*, $k_{\xi\to m}$, we relate it to the conditional probabilities, $P_a(\lambda_{i+1}\,|\,\lambda_i)$, to reach λ_{i+1} from an "active" configuration at λ_i, as follows

$$k_{\xi\to m} \approx \frac{n_{\xi-m}}{n_a^{\xi}} \Phi_{A0} \prod_{i=0}^{\xi-1} P_a(\lambda_{i+1}\,|\,\lambda_i)\,, \quad (21)$$

where $n_{\xi-m}$ is the number of trial runs fired at λ_ξ that started from any point labeled (*a*) and reached *m* before basin A, B, or other intermediate state; $n_a^{\xi} = N_a^{(\xi)} k_\xi$ is the total number of fired trial runs at λ_ξ from (*a*) points, where $N_a^{(\xi)}$ is the number of (*a*) points at λ_ξ. In Eq. (21), $P_a(\lambda_{i+1}\,|\,\lambda_i) = N^{(i+1)} / n_a^i$, where $N^{(i+1)} = N_a^{(i+1)} + N_{\not a}^{(i+1)}$ is the number of trial runs reaching λ_{i+1}.

To evaluate rate constants $k_{A\to *\to m\to B}$, we can use the {*p*,*q*} labels described above and apply Eq. (18) by simply tracing back the TPE pathways of type {*p*,*q*} where *q*=*m* and *p* is any state; i.e., $k_{A\to *\to m\to B} = \Phi_{A0}[n_{A-*-m} / \prod_{i=0}^{n-1} k_i]$, where $n_{A\text{-}*\text{-}m}$ is the total number of such paths. Likewise, for $k_{A\to m\to B}$ and $k_{A\to B}$ one would simply use the paths (from those that reached B) with label {*a*,*m*} and {*a*,*a*}, respectively.

While Eqs. (18) and (21) were derived adapting some of the concepts of the multiple state transition path method of Rogal and Bolhuis,[13] their method allows the calculation of rates for all possible transitions within the phase space including those between intermediate states (like $k_{m_1\to m_2}$) and backwards from B to A. This is not possible from a single FFS simulation which is unidirectional by design.

III. Results

A. Example 1: Lattice protein folding

The protein model adopted here consists of a 48 amino acid sequence that has a unique, compact native structure, whose energy is minimum E_{min}= -20.24 k_BT with NNC = 57 specific or "native" segment-segment contacts. Further details on the model protein, including its sequence, structure, thermodynamics and kinetic behavior are given elsewhere.[19, 20] The folding kinetics and thermodynamic of the system (via FFS-US) were simulated in free space at the corresponding *bulk transition temperature* T_f = 0.27.[19, 20] Recently, we showed that for this simple system, an order parameter model with linear terms for the number of native contacts (NC) and the configurational energy (E) variables:

$$\lambda(NC,E) = p_B(NC,E) = -0.404 + 0.017(NC) - 0.029(E); \qquad (22)$$

is a good estimate for the reaction coordinate that describes the dynamical bottleneck between the folded and unfolded stable states.[11, 12] Furthermore, this model projects the committor probability (p_B) surface on the phase space, i.e., it estimates the probability of any interfacial point stored in the TPE trajectories to commit to the folded state from their NC and E values. Eq. (22) was then used as order parameter to partition the phase space between the unfolded and folded states for the FFS-US simulations, with seven interfaces (n=7) positioned at λ_i ($0 \le i \le n$): $\boldsymbol{\lambda}(x)$ = {0.0,0.1,0.2,0.3,0.5,0.7,0.9,1.0}. The phase space region A was defined by taking $\lambda \le$ 0 and B by $\lambda \ge 1.0$. The calculations were carried out using the DFFS scheme with the number M of trials at each interface fixed to M_i = 100 ($0 \le i < n$) and with starting points randomly sampled from inside the region A.

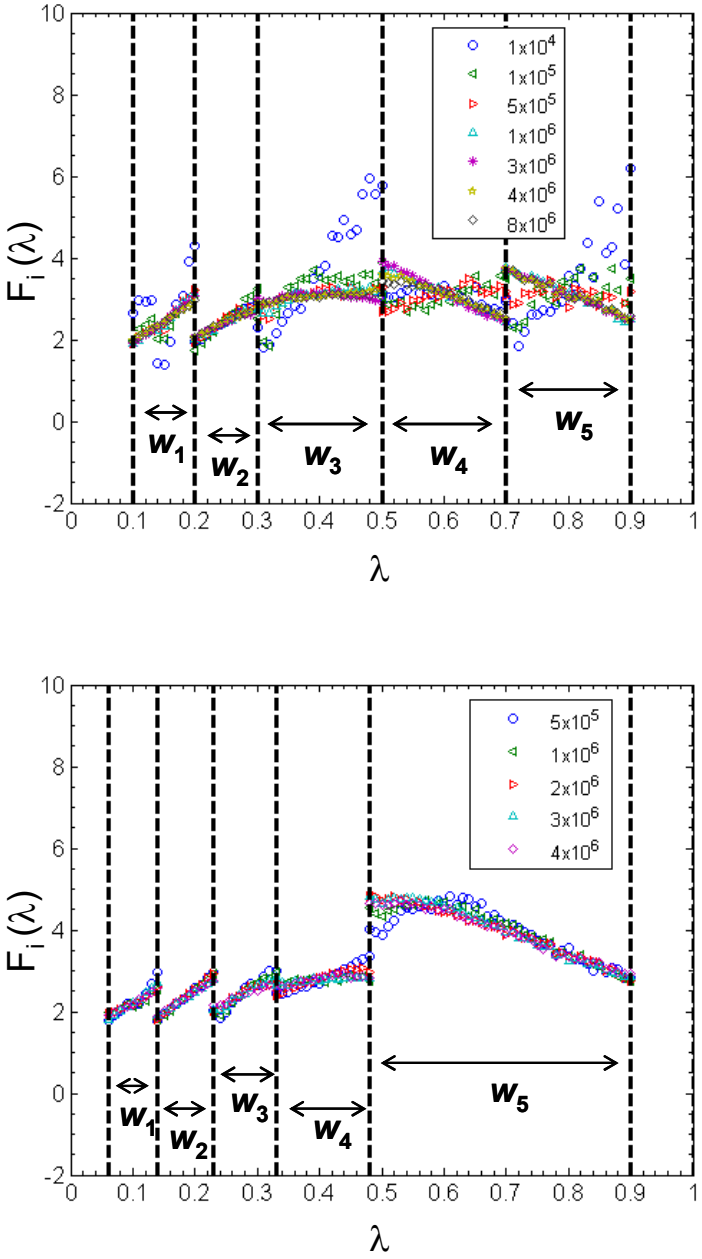

Figure 3.3 $F_i(\lambda)$ functions from standard US simulations of the protein system in different windows for different values of $N_{win}^{(i)}$ and for: (a) un-optimized and (b) optimized λ staging. The thick lines correspond to the windows' walls and the thin lines correspond to the middle interfaces from which the partial paths are initiated.

First, we investigate if our estimate for the maximum number of entries accumulated for each window w_i's histogram ($N_{win}^{(i)}$) [via Eq. (5)] is valid such that enough visits are accumulated for the w_i's $H_i(\lambda)$ to produce the correct $F(\lambda)$ function. For this purpose, we performed a series of conventional umbrella sampling (US) simulations with different $N_{win}^{(i)}$ values for each of the windows shown in Figure 3.3(a). The phase space between the stable states A and B was subdivided into five different window sizes $[\lambda_{i,}\ \lambda_{i+1}]$, overlapping each other by one state. The US simulations were initiated from a randomly selected state at interface λ_i. In Figure 3.3 (a), we compare the $F_i(\lambda)$ functions in different windows obtained with different $N_{win}^{(i)}$ values. Note that the minimum $N_{win}^{(i)}$ value which produce the correct $F_i(\lambda)$ increases with the w_i's size, for example, $N_{win}^{(i)} \sim 10^5$ MC steps for w_1 and w_2 and $N_{win}^{(i)} \sim 4\times10^6$ for $w_{3,}\ w_4$ and w_5 ($w_3 = w_4 = w_5 > w1 = w_2$, where ω stands for the window's width). To verify these results we also consider a subdivision into five windows, resulting from the phase space partition using an optimized λ staging, as shown in Figure 3.3(b). The optimized $\{\lambda'\}$ set was found by an adaptive optimization algorithm,[12] which seeks to allocate the computational effort in a FFS simulation to reduce the statistical error with which the reaction rate constant is estimated. The optimized $\{\lambda'\}$ set corresponds to λ_i ($0 \le i \le n$=7): $\boldsymbol{\lambda}(x)$ = {0.0,0.06,0.14,0.23,0.33,0.48,0.9,1.0}and the region A and B were defined as before by taking $\lambda \le 0$ and $\lambda \ge 1.0$, respectively. Again, the minimum $N_{win}^{(i)}$ value which produces the correct $F_k(\lambda)$ increases with the w_i's size, for example, $N_{win}^{(i)}$ is of order $\sim 10^5$ and 4×10^6 MC steps for w_1 and w_5 ($w_5 > w_1$), respectively. Tables 3.1 and 3.2 give the $N_{win}^{(i)}$ values in each w_i $[\lambda_i,\ \lambda_{i+1}]$ estimated by Eq. (5), where the constant a was calculated for five windows (m=5) and assuming that the total computational time spent by a standard brute force (BF) simulation in that region $[\lambda_1 = 0.06,\ \lambda_6 = 0.9]$ is approximately $t'_{CPU} = 3\times10^7$ MC steps. Eqs. (6) and (7) predicted that the total computational cost to obtain the $p(\lambda)$ distribution from the FFS-

Table 3.1 Data for FFS-US protein folding simulations using an "un-optimized" λ staging, the λ range was subdivided into five US windows w_i. $N_{bin} = 10^4$ and the contribution to each window's $H_i(\lambda)$ from the FFS-type sampling $h_i^{FFS}(\lambda)$ (%R_{FFS}) was determined from 100 DFFS sampling blocks, each one started from a randomly selected point at λ_0.

w_i $[\lambda_i, \lambda_{i+1}]$	λ_i	λ_{i+1}	$N_{win}^{(i)}$ **[MC steps]**	**%** R_{FFS}
1	0.10	0.20	4.21×10^6	62
2	0.20	0.30	4.21×10^6	48
3	0.30	0.50	1.68×10^6	19
4	0.50	0.70	1.68×10^6	39
5	0.70	0.90	1.68×10^6	72

Table 3.2 Data for the FFS-US protein folding simulations using the optimized λ staging, and five US windows w_i. $N_{bin} = 10^4$ and the contribution to each window's $H_i(\lambda)$ from the FFS-type sampling $h_i^{FFS}(\lambda)$ (%R_{FFS}) was determined from 100 DFFS sampling blocks, each one started from a randomly selected point at λ_0.

w_i $[\lambda_i, \lambda_{i+1}]$	λ_i	λ_{i+1}	$N_{win}^{(i)}$ **[MC steps]**	**%** R_{FFS}
1	0.06	0.14	1.7×10^5	52
2	0.14	0.23	2.1×10^5	46
3	0.23	0.33	2.6×10^5	61
4	0.33	0.48	5.9×10^5	50
5	0.48	0.90	4.6×10^6	33

US approach is $\tau \approx 5.9\times10^6$ MC steps, and the constant a is equal to 4.2×10^3 and 2.6×10^3 MC steps for the "un-optimized" and optimized λ staging simulations, respectively. Note in Tables 3.1 and 3.2 that for all w_i the estimated $N_{win}^{(i)}$ values are of the same order as the values predicted by the conventional US simulations in Figure 3.3.

Figures 3.4 (a) and (b) show the $F_i(\lambda)$ functions in different windows obtained with different values of the minimum number of statistics per bin, N_{bin}, for the "un-optimized" and optimized λ staging simulations, respectively. Note that the $F_i(\lambda)$ functions were estimated by accumulating statistics in $H_i(\lambda)$ from several partial paths, each of them allowed to explore the w_i's region until the condition (15) is met. The number of trial paths (M_i) that are sampling the w_i and contribute to $h_i^{FFS}(\lambda)$ was constrained by the condition (14) ($N_{win}^{(i)}$ values are given in Tables 3.1 and 3.2). For example, as expected, the N_{bin} value which produces the correct $F_k(\lambda)$ is of order ~ 5×10^2 and 10^3 MC steps for w_1 and w_5 ($w_5 > w_1$), respectively (see Sec II B). Thus, the contribution to the statistical error (δ) in $p_i(\lambda)$ from just one partial path is of order $\Delta_k \sim O(0.04$ and $0.03)$ for the w_1 and w_5 ($w_5 > w_1$), respectively. Therefore, for this system we assumed that at each w_i a partial trajectory lost its "memory" during a sampling time not much longer than the time required to accomplish condition (15) where N_{bin} is of order ~ 10^4 MC steps.

In Figure 3.5 we show results for the free energy profile obtained from the simulations in each of the five windows in Table 3.2. The free energy in the stable regions A and B was obtained by directly histogramming $p(\lambda)$ by means of two US standard simulations. Note, however, that the histogram $H_A(\lambda)$ for stable state A ($\lambda \le \lambda_0$) could be obtained from the flux term estimation in Eq. (1) (i.e., step one in Sec II B) when the length of the simulation in the basin A is long enough. The continuous free energy profile was then calculated according to Eq. (4) and compared

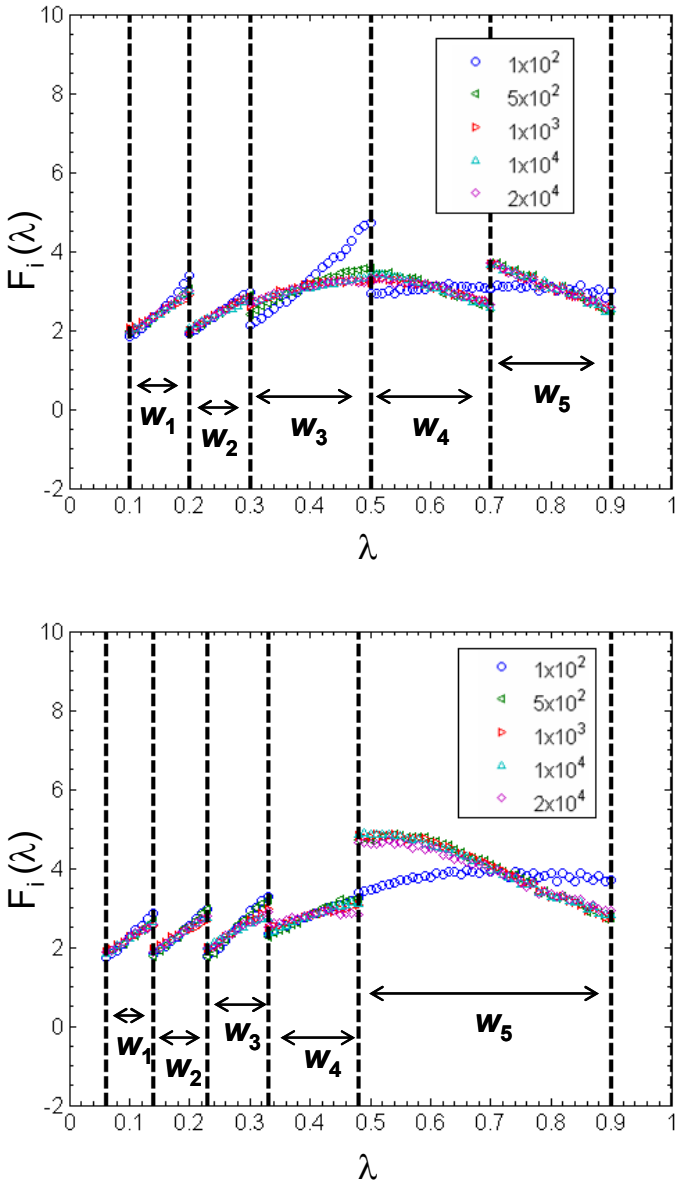

Figure 3.4 $F_i(\lambda)$ functions from FFS-US simulations of the protein system in different windows for different choices of N_{bin} and for: (a) un-optimized and (b) optimized λ staging. The thick lines correspond to the US windows' walls and the thin lines to the middle interfaces from which the partial paths are initiated.

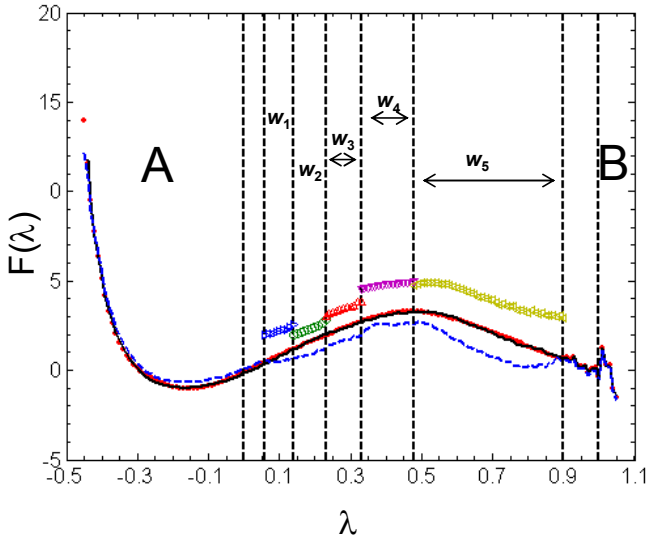

Figure 3.5 Sequence of measured free energies $F(\lambda)$ from the FFS-US simulations for the protein system. The continuous $F(\lambda)$ function was obtained after fitting the results for the $F_i(\lambda)$ in the different windows (●/red). The DFFS scheme was used with an optimized λ staging. The λ staging for the FFS-US simulation is shown by dotted lines. The free energy in the stable regions A and B was obtained by directly histogramming $P(\lambda)$ by means of two standard US simulations. Results from a standard US simulation over the entire domain are also shown (dark line). The free energy profile obtained from the FFS histogram $h_i^{FFS}(\lambda)$ is also shown.

to the results of a standard BF simulation; Figure 3.5 shows that these two profiles agree well. The $F(\lambda)$ function obtained from the FFS histogram $h_i^{FFS}(\lambda)$ is also illustrated in Figure 3.5, showing that the associated $p_i(\lambda)$ distribution is biased. The extra US data corrects for the bias by sampling regions in the phase space not visited by the trajectories contained in the TPE.

As expected, the stage-wise nature of FFS-US leads to a reduction in the computational effort to obtain the $p(\lambda)$ distribution by a factor of m=5 (compared to the cost of a standard BF simulation). Tables 3.1 and 3.2 give values for R_{FFS} from Eq. (16) for the five w_i. Note that, R_{FFS}, the fractional contribution of $h_i^{FFS}(\lambda)$ to the w_i's $H_i(\lambda)$ varies from 20% to 70 %, implying significant savings in computer time.

B. Example 2: Potential Surface 1

At least three main challenges have been identified for the application of FFS to the simulation of rare events:[11, 12] (1) determination of an adequate order parameter (or combination of parameters) that allows the description of multiple transition state regions of a system; (2) assessing high efficiency and completeness of sampling; and (3) estimation of the free energy landscape (i.e., stationary distributions) and barriers. In a previous work,[11] we addressed the first challenge by proposing a new algorithm, FFS-LSE, for identifying suitable reaction coordinates to describe the progression of rare events in complex systems. FFS-LSE uses the transition path ensemble (TPE) obtained from FFS to obtain "on-the-fly" estimates of the committor probability to the final region, p_B, to screen out good order parameter models. More recently,[12] we addressed the second challenge by applying an adaptive algorithm which reduces the statistical error of the estimated transition rate constant (for a given computational cost) by optimizing either the number of trial runs per interface (for fixed staging) or the staging (for fixed interface sampling). We now propose to address the third

challenge by the use of the FFS-US algorithm. In this and the following section, we use all these algorithms to study the diffusion of a particle on a two-dimensional rugged energy surface that is representative of systems with multiple barriers (i.e., metastable states) encountered in many applications.

The model surface under study was used by Chopra et al.[21] and consists of a sum of overlapping Gaussians: 9 major and 100 minor random surfaces. The major surfaces exhibit two well defined global minima at (x,y)=(0.2, 0.2) and (0.8, 0.8) (i.e., A and B stable states, respectively) and three local minima at (0.14, 0.88), (0.35, 0.75) and (0.8, 0.2) which correspond to metastable states. To these major features, 100 random Gaussians are superimposed to increase the roughness of the potential surface. A contour graph of this energy landscape showing the A and B basins is illustrated in Figure 3.6. The initial region was defined by the square region enclosed by $0.1 \le x \le 0.3$ and $0.1 \le y \le 0.3$. The final region was defined by a square region enclosed by $0.7 \le x \le 0.9$ and $0.7 \le y \le 0.9$. The time evolution of the system was simulated using Brownian dynamics at $\beta = 1/k_B T = 0.5$. The parameters that describe the particle diffusion on the surface through Langevin dynamics were: time step $\Delta t = 0.01$, mass m=1.0, and friction coefficient $\gamma = 2.5$. Reflective boundaries were enacted by reflecting the particle moves at the edges with x, y-coordinate < 0 and x, y-coordinate > 1. Figure 3.6 also illustrates the initial $\boldsymbol{\lambda}$ staging: $\lambda_i = x$-coordinate for $0 \le i < n$ was used as initial guess of order parameter. The $\boldsymbol{\lambda}$ space was partitioned into n=9 interfaces positioned at λ_i ($0 \le i < n$): $\boldsymbol{\lambda}(x)$ = {0.31, 0.32, 0.34, 0.36, 0.40, 0.45, 0.50, 0.55, 0.65} and $\lambda_{n=9}$ was taken as the square region enclosing the stable state B. The number of trials per point at λ_i for the BG method was $k_0 = 200$ (i=0) and $k_i = 10$ ($1 \le i < n$).

Figure 3.7(a) shows a map of the probability density (P_{TPE}) of finding a configuration (x, y) in the TPE after a long FFS run [$P_{\mathrm{TPE}}(x, y)$ is incremented by one

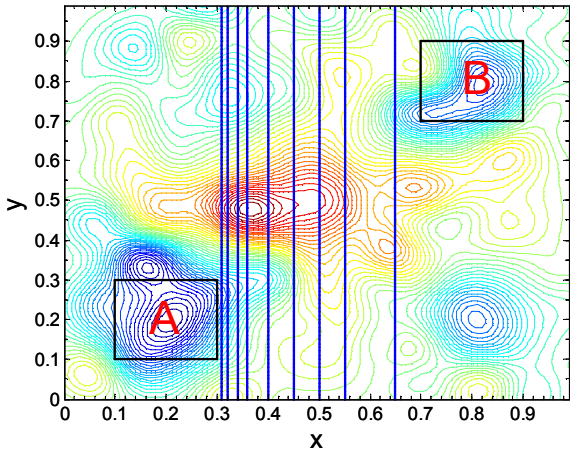

Figure 3.6 Contour graph of the free-energy surface for the two-dimensional potential #1.[21] The color scheme changes from highest (red) to lowest (blue) elevations. The initial and final regions are shown by the squares labeled *A* and *B*, respectively. The initial λ staging for the FFS-type simulation is shown by solid vertical lines (blue).

if a A→B trajectory visits this configuration at least once], where it can be seen that the trajectories connecting the two stable states A and B visit the three metastable states with similar frequency. However, Fig 3.7(b) shows that the free energy profile $F(\lambda=x)$ along this initial guess of order parameter exhibits only two stable states (wells), i.e., the metastable states at the local minima (0.14, 0.88) and (0.35, 0.75) overlap with state A, while the metastable state at (0.8, 0.2) overlaps with state B.

We now explore the possibility that our FFS-LSE method and a staging optimization could help us estimate a good reaction coordinate for this system, allowing a higher computational efficiency for the FFS simulation and the identification of the multiple barriers in the transition. The calculations were carried out with the BG method as a series of blocks, each block consisting of 10 runs, each one started from a randomly selected configuration at λ_0 (out of N_0 = 1000 points generated at λ_0) from which a branched path is generated and used to estimate committor probabilities p_B. The p_B history data is obtained over ten blocks. Following the FFS-LSE protocol, the p_B value for every interfacial point stored in the TPE trajectories was obtained by using recursively Eq. (5) from Ref. 11. Because p_B is the ideal reaction coordinate, a good order parameter model will be one that is able to "fit" well these p_B data.[22, 23] To find such a model, one assumes that p_B follows a mathematical relation (with linear coefficients) that depends on any number of candidate collective properties suspected to be meaningful order parameters; the unknown coefficients are then found by standard least-square estimation (LSE) and the statistically significant terms in the model are found by analysis of variance. The readers are referred to Ref. 11 for more details of the FFS-LSE method. Between blocks, our adaptive optimization algorithm[12] was applied to find a better $\boldsymbol{\lambda}$ staging of the order parameter as follows. The conditional probabilities $P(\lambda_{i+1} \mid \lambda_i)$ to reach subsequent interfaces were tracked to identify "bottlenecks" where sampling should be

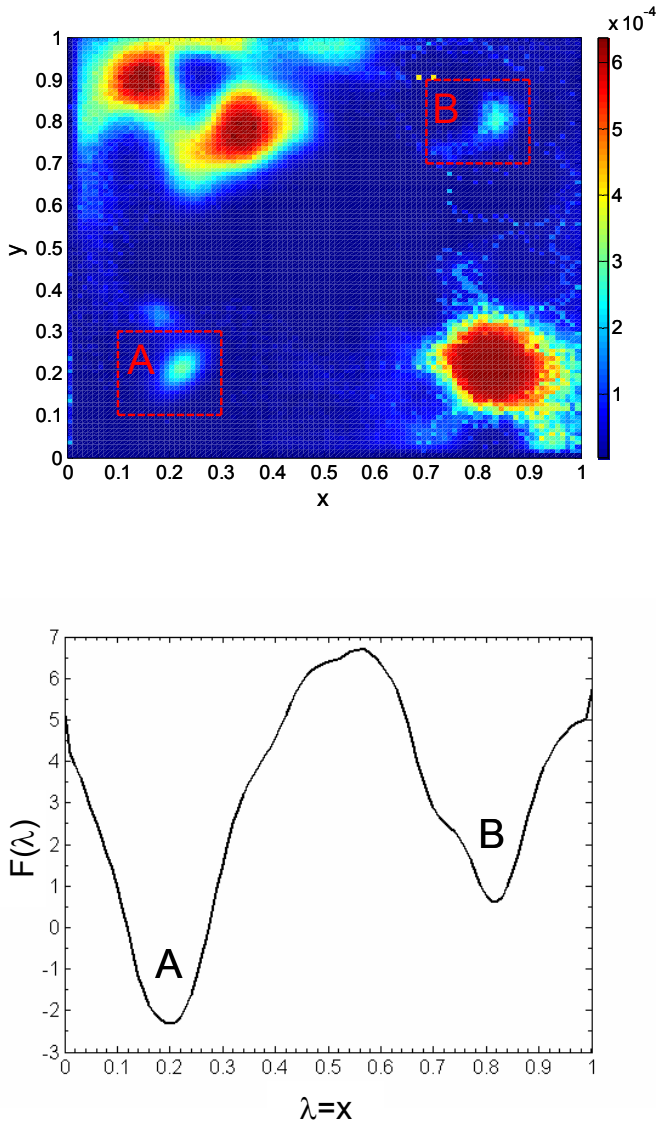

Figure 3.7 Results for the potential energy surface system. (a) Density map (P_{TSE}) obtained from the TPE after a long FFS run. (b) Free energy $F(\lambda=x)$ profile along the x coordinate as reaction coordinate.

Table 3.3 FFS-LSE parameters for the 2D potential energy surface #1 reaction coordinate model [$p_B \approx \beta+\beta_x x+\beta_y y+\beta_{xy} xy+\beta_{xx} x^2+\beta_{yy} y^2+\beta_{xxx} x^3+\beta_{yyy} y^3+\beta_{xxy} x^2 y+\beta_{xyy} xy^2$].

Model	**Model coefficient (β)**									
	constant (β_0)	x	y	xy	x^2	y^2	x^3	y^3	x^2y	xy^2
1	-0.55	1.16	0.72	---	---	---	---	---	---	---
2	-0.11	-0.80	-1.29	-0.37	5.17	5.50	-3.72	-3.67	---	---
3	0.20	-1.68	-1.78	-0.06	6.06	6.25	-4.10	-4.17	---	---

Table 3.4 Data for the FFS-US simulations with energy potential surface #1 and four windows dividing the phase space between stable states. $N_{bin} = 5\times10^3$ and $\%R_{FFS}$ was determined from 10 BG sampling blocks, each one started from ten randomly selected points at λ_0.

w_i $[\lambda_{i-1}, \lambda_{i+1}]$	λ_i	λ_{i+1}	$N_{win}^{(i)}$ $[\Delta t]$	% R_{FFS}
1	0.10	0.18	4.9×10^6	50
2	0.18	0.27	6.3×10^6	15
3	0.27	0.35	4.9×10^6	90
4	0.35	0.90	2.3×10^8	60

concentrated by re-staging. This is done by "interpolating" from the current $P(\lambda_{i+1} \mid \lambda_i)$ vs. λ data [via an auxiliary function defined by Eq. (40) in Ref. 12] to find a new $\{\lambda'\}$ set for which the $P(\lambda_{i+1} \mid \lambda_i)$ values have some pre-specified, desired values [e.g., $P(\lambda_{i+1} \mid \lambda_i) = \left[P(\lambda_n \mid \lambda_0)\right]^{1/n}$ = constant]. The readers are referred to Ref. 12 for a detailed description of the adaptive optimization algorithm. This combination of FFS-LSE and staging optimization provides the advantage of allowing a more efficient and uniform distribution of the p_B data over all the phase space.

Coming back to the present case, the p_B data thus collected were fitted to a tentative regression model that included two collective variables: *x*-, and *y*-coordinate, and interaction terms between these variables. Table 3.3 shows the parameters for reaction coordinate models obtained from the iterations of the combined scheme (i.e., FFS-LSE and adaptive optimization algorithm). For the first iteration, the LSE and analysis of variance for this reaction coordinate model (i.e., model 1 in Table 3.3) indicated that the linear terms for *x*-, and *y*-coordinate are the only significant ones. The p_B surface predicted by this reaction coordinate model (model 1 in Table 3.3) is illustrated in Figure 3.8(a) (dotted lines). A second iteration was performed using this new estimate of order parameter and n=9 interfaces initially positioned at λ_i ($0 \le i < n$): $\boldsymbol{\lambda}(x,y)$ = {0.0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8}. The stables states A and B were defined as before. The number of trials per point at λ_i for the BG method was $k_0 = 200$ (i=0) and $k_i = 10$ ($1 \le i < n$). Note that while we used three iterations to ensure full convergence, two iterations were enough to get suitable convergence. The p_B surface predicted by the reaction coordinate model estimated from the 2nd and 3rd iterations (models 2 and 3 in Table3.3) are illustrated in Figures 3.8(a) (solid lines) and 3.7(b) (dotted lines). The coefficients (and pB surface) in models 2 and 3 are not identical but similar and include significant linear, quadratic and cubic terms for x and y.

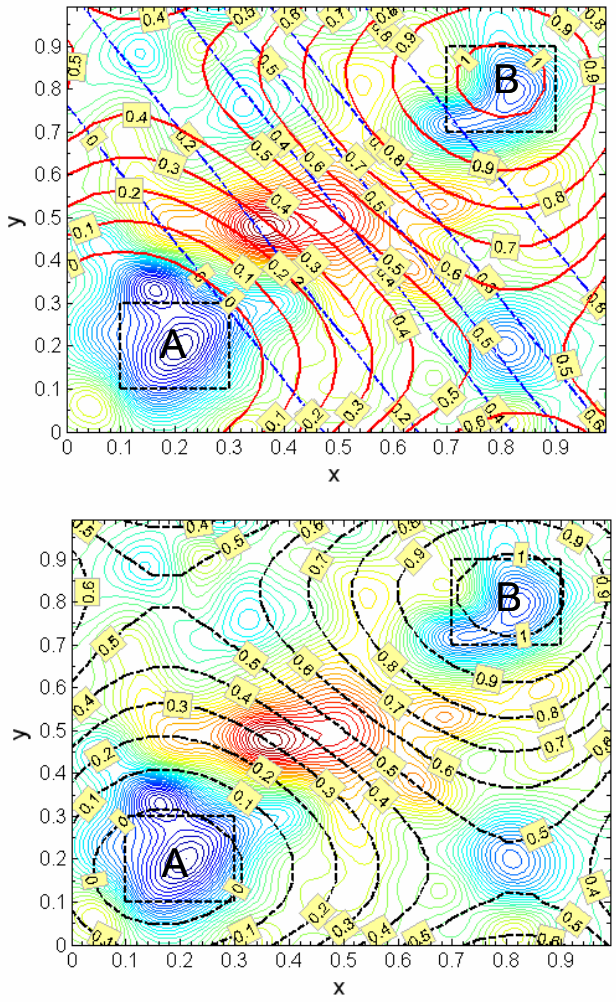

Figure 3.8 Estimated reaction-coordinate iso-lines for the 2D potential energy surface #1. (a) Thin dotted (blue) and solid (red) lines are the predicted p_B committors from model 1 and 2 in Table 3.3, respectively. (b) The dotted lines (black) are the predicted p_B committors from model 3 in Table 3.3. The committor values are shown as labels. Contour plot of the free energy landscape is shown as background for visual reference and the color scheme changes from highest (red) to lowest (blue) elevations. The initial and final regions are shown by the squares labeled *A* and *B*, respectively.

Model 3 was then used as order parameter to partition the phase space between the A and B states for the FFS-US simulations; with eight interfaces (n=8) positioned at λ_i ($0 \le i \le n$): $\boldsymbol{\lambda}(x)$ = {0.0,0.05,0.07,0.10,0.18,0.27,0.35,0.9,1.0}. Region A was defined by taking $\lambda \le 0$ and region B by $\lambda \ge 1.0$. Note that this λ staging corresponds to the optimized $\{\lambda_i'\}$ set obtained by distributing the $P(\lambda_{i+1} \mid \lambda_i)$ values $P(\lambda_{i+1} \mid \lambda_i) = P_\lambda$ for $0 \le \lambda_i \le 6$, where $P_\lambda = \left[P(\lambda_{n=B} \mid \lambda_0) / \prod_{i=1}^{n-1} P(\lambda_{i+1} \mid \lambda_i)\right]^{1/7} \approx 0.31$ and $P(\lambda_8 \mid \lambda_7) = 0.9$ for λ_7; this guarantees that that the 7th interface is placed in the region close to the basin of attraction B and that w_7 (i.e., the last window) encloses the entire phase space region between the transition state (TS) and the region close to basin B ($\lambda_{n=8(B)} \ge 1.0$), as seen in Figure 3.9. The calculations were carried out using the BG scheme with k_0 = 200 (i=0) and k_i = 10 ($1 \le i < n$). Table 3.4 gives the w_i's size and its corresponding $N_{win}^{(i)}$ values estimated by Eq. (5), where the constant a was estimated for four windows (m=4) and assuming that the total computational time required by a standard BF simulation in that region [λ_3 = 0.10, λ_7 = 0.90] is approximately t'_{CPU} = $1\times10^9\,\Delta t$. Eqs. (6) and (7) predicted that the total computational cost to obtain the $p(\lambda)$ distribution from the FFS-US approach is $\tau \approx 2.5\times10^8\ \Delta t$, and that the constant a is ~ $7.7\times10^4\ \Delta t$. The continuous free energy $F(\lambda)$ profile along the reaction coordinate model 3 using a standard BF simulation of length $2\times10^9\ \Delta t$ is shown in Figure 3.9. The resulting λ staging and the windowing of the phase space for the FFS-US simulation are also illustrated in Figure 3.9. Excellent agreement is obtained between the results of the FFS-US and BF calculations. The free energy in the stable regions A and B were obtained by directly histogramming $H_A(\lambda)$ for ($\lambda \le \lambda_1$) and $H_B(\lambda)$ for ($\lambda \ge \lambda_7$) during the flux term estimation in Eq. (1) and a standard BF simulations in the region B, respectively. Note that the $F(\lambda)$ profile along the reaction coordinate model 3 now shows the two stable states A and B separated by a local minimum at $\lambda(x, y) = p_B = 0.5$ and two barriers of similar height; this local minimum pools

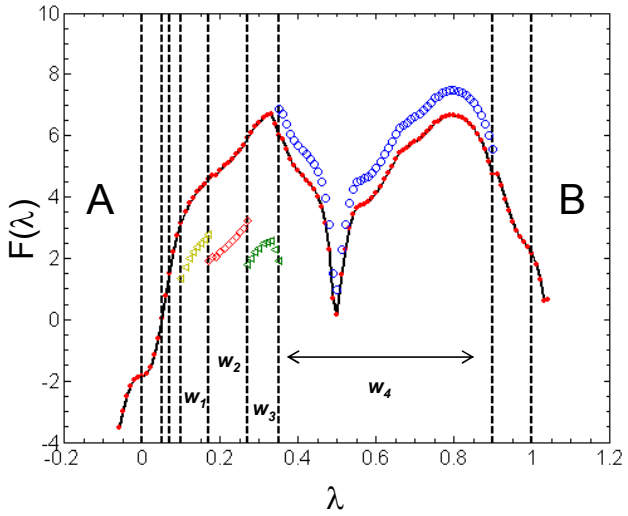

Figure 3.9 Sequence of measured free energies $F(\lambda)$ along reaction coordinate model 3 (see Table IV) from the FFS-US simulations for the 2D potential energy surface #1. The FFS-US λ windows are demarked by dotted lines and the partial $F(\lambda)$ curves in those windows are stitched up together into a continuous $F(\lambda)$ function shown (●/red). The free energy of the initial and final regions (labeled *A* and *B*, respectively) was obtained via standard US simulations. The $F(\lambda)$ function from a standard brute force simulation is also shown (solid/black line).

together (in 1D) the three metastable states of the 2D-surface. In getting $p(\lambda)$, the computational effort is reduced ~m=4 fold due to staging (compared to that of a standard BF simulation), and $h_i^{FFS}(\lambda)$ contributes from 15% to 90% of the data (see $\%R_{FFS}$ in Table 3.4).

Although the multiple state FFS algorithm of Sec. II E could be used here to estimate rate constants for some of the intermediate transitions, we restrict such calculations to the following example for which our estimates can be directly compared with previous literature results.

C. Example 3: Potential Surface 2

In this section, we also study the diffusion of a particle on a two-dimensional energy surface with multiple metastable states. In contract to the previous example (Sec. II B), the trajectories contained on the TPE and connecting both stable states A and B in phase space visit two intermediate states with different frequency, and the surface outside the stable and metastable states is smooth rather than rugged. Besides showing the validity of FFS-US, this example will show that the FFS-LSE derived reaction coordinate allows the identification of the multiple intermediate states in the transition region and the sampling of pathways that connect any two stable or intermediate states (from which rate constants of all possible transitions can be estimated). The model surface is described in detail in Ref. 13. Figure 3.10(a) shows a contour graph of the energy landscape showing the two well defined global minima at (x,y)=(-4, 0) and (4, 0) (i.e., stable states A and B, respectively) and two local minima at (-3, 4.8) and (-0.5, 3.2) (i.e., metastable states I and II, respectively). The four minima basins are enclosed by circles around the minima with a radius r_i set to a r_A = r_B = 1.0 and r_I = r_{II} = 0.5. The time evolution of the system was simulated using Brownian dynamics at $\beta = 1/k_B T = 2.5$. The parameters that describe the particle

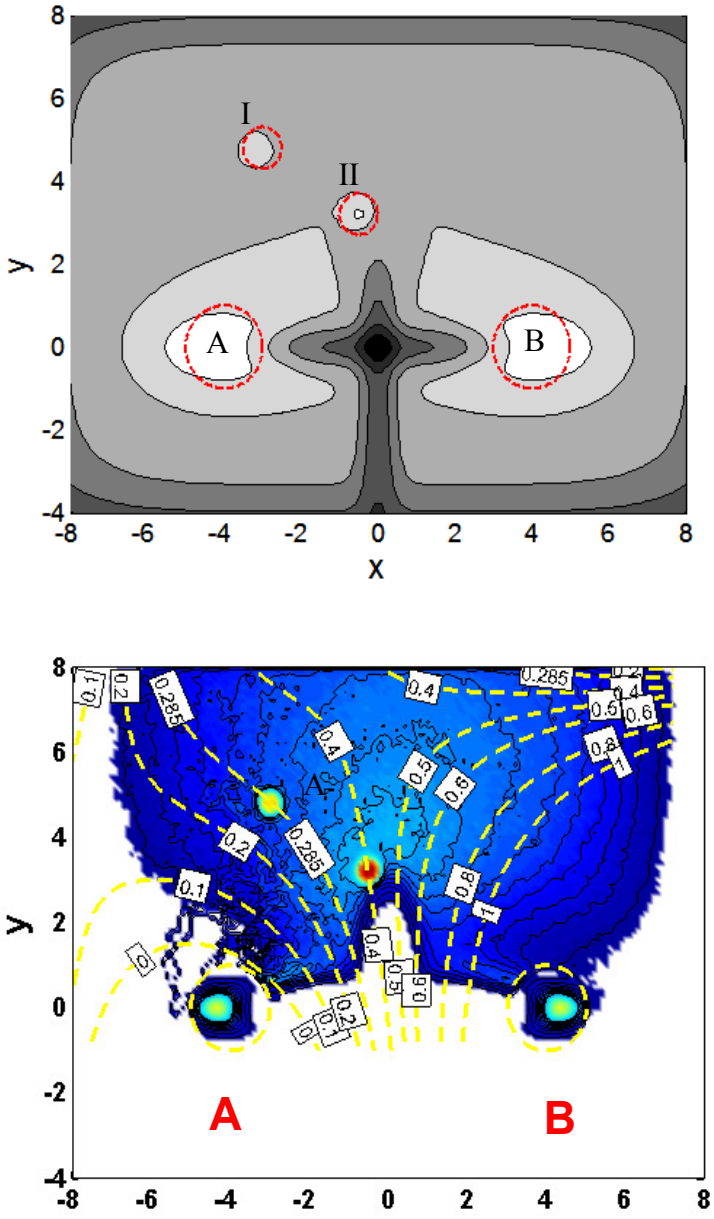

Figure 3.10 (a) Contour graph of the free-energy surface for the two-dimensional potential #2.[13] The color scheme changes from highest (gray) to lowest (white) elevations. The basins are shown by the cycles labeled A, B, I, and II. (b) Density map (P_{TSE}) obtained from the TPE after a long FFS run and the predicted p_B committors (dotted lines, yellow) from the reaction coordinate model found by FFS-LSE; committor values are shown as labels. The color scheme changes from highest (red) to lowest (blue) densities. The initial and final regions are shown by the circles labeled *A* and *B*, respectively.

diffusion on the surface through Langevin dynamics were: time step $\Delta t = 0.1$, mass m=1.0, and friction coefficient $\gamma = 2.5$. Reflective boundaries were enacted to keep the particle inside -8<x<8 and -4<y<8.

Figure 3.10(b) shows a map of the probability density [P_{TPE}(x, y)] of finding a configuration (x, y) in the TPE after a long FFS run, where it can be seen that the trajectories connecting the two stable states A and B visit the two metastable states with different frequency. Fig 3.11(a) shows that the free energy profile $F(\lambda=x)$ along the x-coordinate as order parameter exhibits only three wells: the stable states A and B, and the intermediate state II; the metastable states I is not well defined along this order parameter. To implement the FFS-LSE method, we used the x-coordinate as initial guess of order parameter, the $\boldsymbol{\lambda}$ space was partitioned into n=9 interfaces positioned at λ_i ($0 \le i < n$-1): $\boldsymbol{\lambda}(x)$ = {-3,-2.5,-2,-1.5,-1,-0.5,0,1,2}, and $\lambda_{n=9}$ was taken as the circular region enclosing state B. For the BG runs, $k_0 = 200$ (i=0), $k_i = 10$ ($1 \le i < n$) and the calculations were carried out as a series of blocks, each one consisting of 10 BG runs. Between blocks, our adaptive optimization algorithm[12] was applied to find a better $\boldsymbol{\lambda}$ staging of the order parameter as described in the Sec. II B. Figure 3.10(b) also shows isosurfaces for the reaction coordinate obtained from the p_B data collected after two iterations of the combined scheme and fitted to a tentative regression model that included the x, and y coordinates and their quadratic and cubic interaction terms; the resulting model is $p_B \approx 0.4151 + 0.2402\,x + 0.0287\,y - 0.0330\,xy + 0.0289\,x^2 - 0.0039\,y^2 + 0.0005\,x^3 - 0.0042\,x^2y$.

The optimized order parameter model was then used to partition the phase space between the A and B states for the FFS-US simulations; with nine interfaces (n=9) positioned at λ_i ($0 \le i \le n$-1): $\lambda(x)$ = {0.0, 0.05, 0.10, 0.20, 0.30, 0.40, 0.50, 0.60, 0.9}. Region A and B were defined by circular regions enclosing the stable A

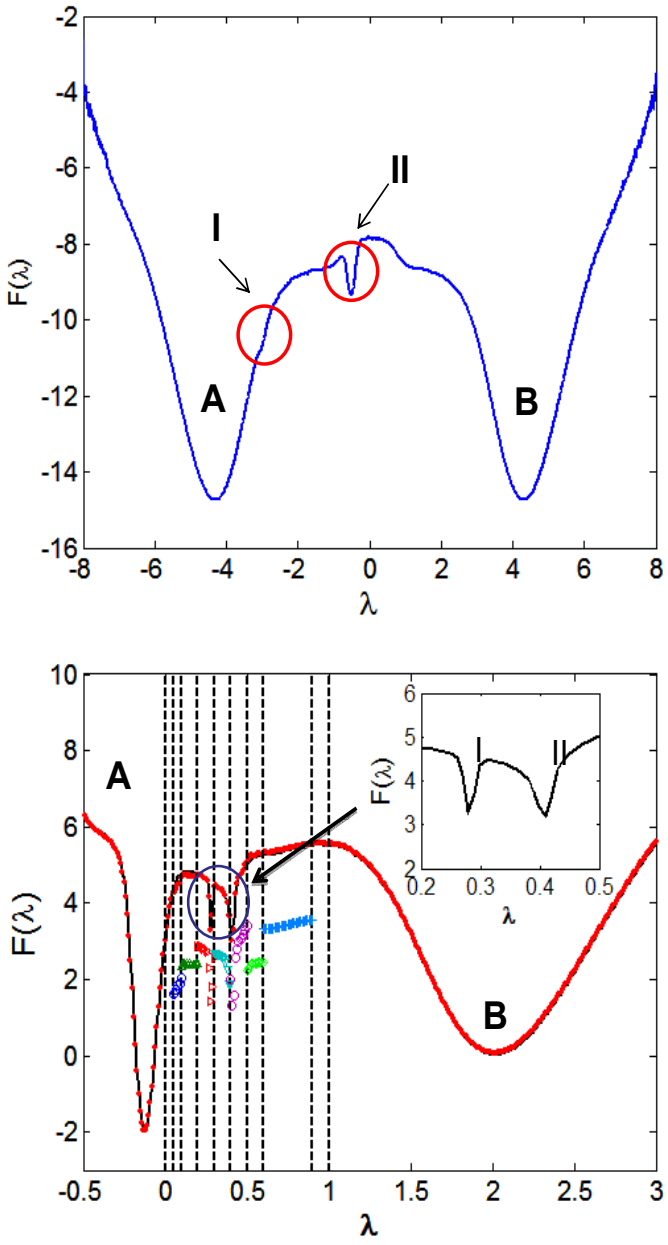

Figure 3.11 Free energy F for the 2D potential energy surface #2. (a) F profile along the x coordinate as reaction coordinate. (b) Sequence of measured free energies $F(\lambda)$ along optimized reaction coordinate model ($\lambda=p_B$) from the FFS-US simulations. The λ windows are demarked by dotted lines and the partial $F(\lambda)$ curves in those windows are stitched up into a continuous $F(\lambda)$ function (●/red). $F(\lambda)$ in the intial region A and final region B was obtained via standard US simulations. $F(\lambda)$ from a standard brute force simulation is also shown (solid/black line).

and B minima, respectively. The calculations were carried out using the BG scheme with k_0 = 200 (i=0) and k_i = 10 ($1 \le i < n$). Table 3.5 gives the w_i's size and its corresponding $N_{win}^{(i)}$ values estimated by Eq. (5), where the constant a was estimated for seven windows (m=7) and assuming that the total computational time required by a standard BF simulation in that region [$\lambda_1 = 0.05, \lambda_8 = 0.90$] is approximately $t'_{CPU} = 10^9\ \Delta t$. Eqs. (6) and (7) predicted that $\tau \approx 1.4\times10^7\ \Delta t$, and a is $\sim 9.8\times10^3\ \Delta t$. The continuous free energy $F(\lambda)$ profile along the optimum reaction coordinate model using a standard BF simulation of length $2\times10^9\ \Delta t$ is shown in Figure 3.11(b). The resulting λ staging and the windowing of the phase space for the FFS-US simulation are also illustrated in Figure 3.11(b). Excellent agreement is obtained between the results of the FFS-US and BF calculations. The free energy in the stable regions A and B were obtained by directly histogramming $H_A(\lambda)$ for $(\lambda \le \lambda_1)$ and $H_B(\lambda)$ for $(\lambda \ge \lambda_8)$ during the flux term estimation in Eq. (1) and a standard BF simulation in the region B, respectively. Note in Figure 3.11(b) that the $F(\lambda)$ profile along the optimized reaction coordinate model now shows the two states A and B separated by two local minima at $\lambda(x, y) = p_B = 0.285$ and 0.40 for intermediates states I and II, respectively. In obtaining the $p(\lambda)$ distribution, FFS-US capitalizes on the $h_i^{FFS}(\lambda)$ data which contributed between 38% to 95% in terms of $\%R_{FFS}$ (see Table 3.5).

The optimized reaction coordinate model shows that particles visiting the intermediate state II have greater probability to commit to region B compared to particles visiting state I. Consequently, the rate constant for the paths passing through II (i.e., A→II→B and A→∗→II→B) is expected to be higher than those for transitions passing through I (i.e., A→I→B and A→ ∗→I→B); see Sec. II E for the notation details. Consistent with the density map (P_{TSE}) in Figure 3.10(b), the TPE should contain more pathways connecting the stable states through state II than through state

Table 3.5 Data for the FFS-US simulations with the energy potential surface #2 and seven windows w_i between states A and B. $N_{bin} = 5\times10^3$.

w_i $[\lambda_{i-1}, \lambda_{i+1}]$	λ_i	λ_{i+1}	$N_{win}^{(i)}$ $[\Delta t]$	% R_{FFS}
1	0.05	0.10	2.5×10^5	38
2	0.10	0.20	9.8×10^5	38
3	0.20	0.30	9.8×10^5	39
4	0.30	0.40	9.8×10^5	52
5	0.40	0.50	9.8×10^5	83
6	0.50	0.60	9.8×10^5	95
7	0.60	0.90	8.9×10^6	89

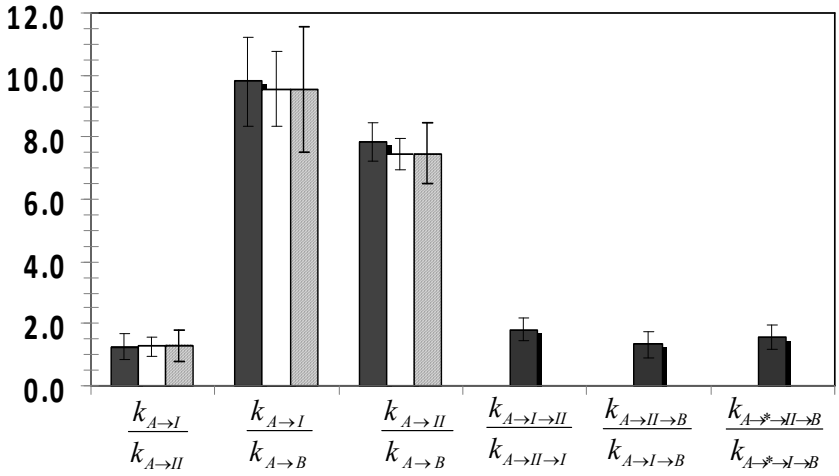

Figure 3.12 Schematic illustration of the reactions involved in the A to B transition for the 2D potential energy surface #2.[13] (b) Ratio of the multiple state probabilities (i.e., k_{Ai}/k_{Aj}) for different transitions between the four basins: from a single multiple state FFS simulation (black bars), independent FFS simulations (white bars) and from Figure 3 of Ref. 13 (patterned bars). From a single FFS simulation connecting stable states A and B, rate constant values can be estimated for the transitions $k_{A\to B}, k_{A\to I\to B}, k_{A\to II\to B}, k_{A\to *\to I\to B}, k_{A\to *II\to B}, k_{A\to I}$, $k_{A\to II}, k_{A\to I\to II}, k_{A\to II\to I}$ and $k^{T}_{A\to B}$.

I. Figure 3.12(a) shows a schematic illustration for all the possible transitions connecting stable states A and B. The rate constant estimates for transition from state A to B and passing through M metastable states **m** (i.e., $M \geq 1$) and from state A to a particular m state were estimated by Eq. (18) and (20), respectively. The results in Figure 3.12(b) for the ratio of the multiple transition rate constants show good agreement with the results presented in Figure 3 of Ref. 13 and those we obtained (as a further test) from independent FFS simulations targeting a particular transition. The rate constants for transitions from A→I and A→II are similar in magnitude whereas the rate constant A→B ($k_{A\to B}$) for trajectories connecting A and B without passing through an intermediate state is a factor of ~ 7 lower. Furthermore, transitions to the intermediate state II through state I (A→I →II) are a factor of ~ 2 larger than those passing first through state II before committing to sate I (A→II→I). Hence, as expected from the p_B surface, once that the system committed to the intermediate state II, it is most likely to make a transition to the basin of attraction B. The rate for transitions through intermediate state II (A→*→II→B) are a factor of ~ 1.5 larger than those through state I (A→*→I→B) and a factor 3.3 larger than those that go directly from A to B (A→B), indicating that most paths arriving to B come from state II. Moreover, transitions through intermediate state II only (A→II→B) are a factor of ~ 1.3 larger than those through state I only (A→I→B). The rates for the A→II→B and A→I→B transitions are 63 and 74 % of those for the A→*→II→B and A→*→I→B transitions, respectively, indicating that the amount of I↔II transitions is significant.

IV. Conclusions

This paper complements two previous articles[11, 12] in which we presented several optimization algorithms for FFS-type simulations. The new algorithm (FFS-US) uses the FFS setup to obtain the equilibrium probability distribution once the

kinetics of the system was simulated via FFS. The scheme is straightforward to implement and does not involve the use of biasing weights. The correct probability distribution is computed by combining histogram data from both the transition path ensemble obtained from FFS and an extra US performed for several partial paths. By design, the computational effort is set to be the same as that of a standard US simulation so as to have comparable statistical errors in the free energy data. Our proof-of principle applications demonstrate the validity of the FFS-US for computing the equilibrium distribution of systems involving free-energy barriers having multiple local-minima.

In FFS-US, the FFS runs typically provided more than half of the total statistical data needed to evaluate the probability distributions, with US providing the balance. Perhaps more importantly, the initial interfacial points provided by FFS are ideal for the subsequent US because they are obtained along an optimized order parameter (which should help minimize hysteretic effects) with windows positioned for computational efficiency. In this manner, the resulting free-energy is mapped along a most useful coordinate while concentrating the US in the more difficult/important regions. Also, FFS-US should be helpful in reducing window kinetic trapping which tends to occur with conventional window-based US methods: by sampling windows using several paths initiated at different configurations (inherited from FFS) of a given interface, the system is allowed to more broadly explore the phase space around the barriers.

We also extended the original FFS formalism so that sampling of pathways passing through multiple intermediate states can be monitored to allow the estimation of rate constants of all possible forward transitions connecting stable states A and B through intermediates states and from A to any intermediate state. We envision that this method will be a valuable tool to study complex systems where the transition

between the two main basin is mediated by partial transitions through distinct metastable states.

Acknowledgments

The authors are grateful for support from the National Science Foundation Award 0553719.

REFERENCES

1. Allen, R. J.; Frenkel, D.; ten Wolde, P. R., *J. Chem. Phys.* **2006,** 124, 024102.

2. Allen, R. J.; Warren, P. B.; ten Wolde, P. R., *Phys. Rev. Lett.* **2005,** 94, 018104.

3. Frenkel, D.; Smit, B., *Understanding Molecular Simulation: From Algorithms to Applications*. 2nd ed.; Academic: Boston, 2002.

4. Radhakrishnan, R.; Schlick, T., *J. Chem. Phys.* **2004,** 121, 2436-2444.

5. Moroni, D.; van Erp, T. S.; Bolhuis, P. G., *Phys. Rev. E* **2005,** 71.

6. Faradjian, A. K.; Elber, R., *J. Chem. Phys.* **2004,** 120, 10880-10889.

7. Ytreberg, F. M.; Zuckerman, D. M., *Proc. Nat. Acad. Sci. U.S.A.* **2008,** 105, 7982-7987.

8. Valeriani, C.; Allen, R. J.; Morelli, M. J.; Frenkel, D.; ten Wolde, P. R., *J. Chem. Phys.* **2007,** 127.

9. Bolhuis, P. G., *J. Chem. Phys.* **2008,** 129, 114108-13.

10. Warmflash, A.; Bhimalapuram, P.; Dinner, A. R., *J. Chem. Phys.* **2007,** 127, 154112-8.

11. Borrero, E. E.; Escobedo, F. A., *J. Chem. Phys.* **2007,** 127, 164101-17.

12. Borrero, E. E.; Escobedo, F. A., *J. Chem. Phys.* **2008,** 129, 024115-16.

13. Rogal, J.; Bolhuis, P. G., *J. Chem. Phys.* **2008,** 129.

14. Allen, R. J.; Frenkel, D.; ten Wolde, P. R., *J. Chem. Phys.* **2006,** 124, 194111.

15. Moroni, D.; van Erp, T. S.; Bolhuis, P. G., *Physica A (Amsterdam)* **2004,** 340, 395-401.

16. Virnau, P.; Muller, M., *J. Chem. Phys.* **2004,** 120, 10925-10930.

17. Ferrenberg, A. M.; Swendsen, R. H., *Phys. Rev. Lett.* **1989,** 63, 1195-1198.

18. Escobedo, F. A.; Abreu, C. R. A., *J. Chem. Phys.* **2006,** 124.

19. Abkevich, V. I.; Gutin, A. M.; Shakhnovich, E. I., *Fold Des.* **1996,** 13, 221-230.

20. Borrero, E. E.; Escobedo, F. A., *J. Chem. Phys.* **2006,** 125, 164904-14.

21. Chopra, M.; Malshe, R.; Reddy, A. S.; de Pablo, J. J., *J. Chem. Phys.* **2008,** 128.

22. Ma, A.; Dinner, A. R., *J. Phys. Chem. B* **2005,** 109, 6769-6779.

23. Peters, B.; Trout, B. L., *J. Chem. Phys.* **2006,** 125.

4. CHAPTER: Folding kinetics of a lattice protein via a forward flux sampling approach*

Abstract

We implement a forward flux sampling approach [R. J. Allen, D. Frenkel, and P. R. ten Wolde, J. Chem. Phys. 124, 194111 (2006)] for calculating transition rate constants and for sampling paths of protein folding events. The algorithm generates trajectories for the transition between the unfolded and folded states as chains of partially connected paths, which can be used to obtain the transition state ensemble and the properties that characterize these intermediates. We apply this approach to Monte Carlo simulations of a model lattice protein in open space and in confined spaces of varying dimensions. We study the effect of confinement on both protein thermodynamic stability and folding kinetics; the former by mapping free energy landscapes and the latter by the determination of rate constants and mechanistic details of the folding pathway. Our results show that, for the range of temperatures where the native state is stable, confinement of a protein destabilizes the unfolded state by reducing its entropy, resulting in increased thermodynamic stability of the folded state. Relative to the folding in open space, we find that the kinetics can be accelerated at temperatures above the temperature at which the unconfined protein folds fastest and that the rate constant increases with the number of constrained dimensions. By examining the statistical properties of the transition state ensemble, we detect signs of a classical nucleation folding mechanism for a core of native contacts formed at an early stage of the process. This nucleus acts as folding foci and is composed of those

---

* E. E. Borrero and F. A. Escobedo, J. Chem. Phys. 125, 164904 (2006).

residues that have higher probability to form native contacts in the transition state intermediates, which can vary depending on the confinement conditions of the system.

I. Introduction

The understanding of how a protein folds successfully into its native structure capable of complex biological activity is key in the development of bio-therapeutics and in the analysis of debilitating human diseases (such as Alzheimer's, Huntington's, and Parkinson's) that are associated with protein misfolding and aggregation.[1,2] In addition to the interior of the ribosomal tunnel, which provides the first type of encaging a polypeptide chain encounters during folding, [3] other natural occurrences of spatial confinement found in living cells include chaperon cages that have evolved to play key roles in protein folding and in the prevention of protein misfolding.[4-6] Numerous simulation studies have been performed to explain the effect of protein-surface interactions and chaperon-assisted protein folding and to show that confinement alone can induce changes in the way amino acids interact through the folding pathway.[3,7-11] Coarse-grained lattice models with dynamic Monte Carlo (MC) methods have proven to be a useful computational tool to get insights of global properties and the mechanism of folding in open and confined space. [7,12] Other researchers have applied molecular dynamics to simulate crowding inside the cellular space and chaperonin confinement by restricting the folding process of the protein to a spherical or cylindrical cage.[7,9,10] Most of these simulation studies have concluded that surface-protein interactions restrict the conformational space explored by the protein (during folding) due to excluded volume and/or chemical interactions leading to changes in the folding rate and mechanism; these changes can make the native protein structure more stable and the folding process faster. However, these studies typically lack an analysis of the intermediates observed during folding and of the properties of

the transition state ensemble. Moreover, most of these simulation studies characterized the folding kinetics by determining the average or median folding time (τ_f) from simulations initiated from different structures chosen randomly from unfolded conformations generated at high temperatures. To support predictions from thermodynamic calculations and to elucidate how surface-protein interaction could affect the mechanism and dynamics of the folding pathways, it is essential to be able to calculate rate constants and to sample transitions paths for rare events.

For many natural rare events in biological systems, including protein folding, protein aggregation, transport-driven through membranes, and adsorption, the kinetics (i.e., rate constants and mechanism) of the system is often very difficult to simulate by conventional "brute-force" approaches.[13,14] This is because the average waiting time between events is orders of magnitude longer than the accessible simulation time itself, and the vast majority of the computational effort is spent in simulating the uninteresting waiting period between events. Even when coarse-grained protein lattice models with dynamic MC simulations are used, the required number of independent simulations needed to obtain rate constants values with acceptable uncertainty is extremely large, as shown later in Sec. IV B. Furthermore, from these simulations, it is difficult to obtain mechanism and details of the folding pathways and to collect the transition path ensemble (TPE) between the unfolded and folded states and the properties of the transition state ensemble (TSE). Simulation of the kinetics of stochastic events like protein folding requires the use of specialized techniques such as the Bennett-Chandler method, [15,16] the Crooks and Chandler approach,[17] transition path sampling (TPS),[18-21] milestoning methods,[22] transition interface sampling (TIS),[23,24] and partial path transition interface sampling (PPTIS).[25] The Bennett-Chandler approach is based on transition-state theory and the reaction rate is estimated via the calculation of the free energy barrier along a predetermined reaction

coordinate.[16] This method does not provide information about the transition trajectories, and the choice of an inappropriate coordinate leads to inefficient calculation of the rate constant (which is especially troublesome for high-dimensional complex systems). Other methods, such as TPS have been developed to generate transition paths that overcome the problem of reaction coordinate selection.[21] TPS is based on an algorithm that calculates the probability for every possible trajectory of a particular duration in systems exhibiting stable states; this leads to the construction of partitions functions for ensembles of trajectories. The TPE is sampled using a Monte Carlo procedure that generates new paths by "shooting" forward and backward in time from already existing paths. Its has shown that the TPE thus generated provides a viable method for obtaining mechanistic details and rate constants for protein folding processes.[18] However, TPS requires knowledge of the initial state phase density, which means that the system must be in equilibrium. Moreover, TPS requires that the transition between states along the path occurs rapidly, since new paths are generated by shooting and shifting moves that tend to diverge, leading to inefficient sampling.[19] If these criteria are not met, TPS may lead to the collection of non-reactive trajectories. Others methods, like the Crooks and Chandler approach, can be applied to nonequilibrium systems and rely on a methodology that resembles the TPS in generating new paths from old paths by changing the random number history.[17] In TIS[24] and PPTIS,[25] the TPE is sampled via a series of interfaces in phase space to facilitate the generation of transition paths by a TPS-like procedure. These methods assume Markovian "memory loss" over subsequent interfaces. Milestoning methods generate trajectories between interfaces assuming a steady-state distribution at each interface (i.e., probabilities do not depend on the history of the paths), an assumption which is unlikely to be justified in many systems.[22]

Recently, new alternative algorithms have been developed that do not require the knowledge of the phase-space density and can therefore be used for nonequilibrium systems with stochastic dynamics. In addition, these sampling schemes avoid many of the difficulties associated with existing equilibrium rare-event methods and do not require the specification of a reaction coordinate. These new algorithms are known as "forward flux sampling" (FFS)-type simulation schemes.[14] They allow the computation of both rate constants and TPE by dividing the phase space between the initial and final region into a series of interfaces. The TPE is obtained by connecting sequentially generated partial paths between the interfaces. Moreover, in these methods trajectories are obtained without any requirement on their length. The rate constant for the stochastic system is given by the product of the flux of trajectories crossing the first interface and the probability that these trajectories reach the final state. Although a series of interfaces in the phase-space is used as in other sampling techniques (e.g., TIS and PPTIS), no assumption is made about the distribution of paths at the interfaces;[14] i.e., trajectories are generated without assuming a steady-state distribution (or "memory loss" during the transition) at each interface.

In this article, we explored the ability of a (FFS)-type path sampling scheme to obtain rate constants and mechanistic details for the folding of a model protein. First, we show that it is possible to apply the FFS-type method to a simple cubic lattice protein model and obtain rate constants, the TPE, and properties of the TSE for the folding pathway with enhanced efficiency compared to brute-force and TPS simulations techniques. Second, we demonstrated that the folding mechanism and folding-rate constants obtained are consistent with predictions from free-energy calculations for proteins inside a confined space restricted by non-interactive walls. Our results are in accordance with previous ones, in which averages for the first

passage folding time were used to describe the kinetics of the system. However, we also obtain mechanistic details of the intermediates and properties of the transition state of the process.

In Sec. II, we give a detailed description of (FFS)-type simulation schemes for the calculation of rate constants and transition pathways (Sec. II A and B), and the analysis of the TPE to obtain information on the mechanism by which the rare event occurs (Sec. II C). In Sec. III, we present the models and methods used for the protein folding of a lattice protein. In Sec. IV, we discuss and compare the results from free energy calculations and kinetic simulations. In Sec. IV B, we also compare kinetic results obtained from brute force calculation, FFS-type, and TPS schemes for the same system. In Sec. V we provide some concluding remarks and discuss their relevance to previous literature results. Finally, in the Appendix we give a brief description of the TPS theory and the calculation of rate constants for the lattice protein.

II. FFS-Type Sampling Scheme

A. Theoretical background

The FFS-type algorithms generate trajectories for rare events between two well-defined states A and B (e.g., unfolded and folded) of a system. Regions A and B are defined in terms of an order parameter $\lambda(x)$, where x denotes the coordinates of the phase space. The parameter $\lambda(x)$ can be chosen such that the system has values $\lambda(x) \leq \lambda_A(x)$ in region A and $\lambda(x) \geq \lambda_B(x)$ in region B. The rate constant of the process is estimated as an average rate of transitions from A to B using an "effective positive flux" expression.[13,14,26] To this end, it employs a series of nonintersecting interfaces (n+1) between the initial (A) and final (B) regions $\{\lambda_0,\ldots\ldots,\lambda_n\}$, such that $\lambda \geq \lambda_0$, $\lambda_n = \lambda_B$, and $\lambda_i > \lambda_{i-1}$ (λ increases monotonically as the interfaces $\lambda_0,\ldots,\lambda_n$ are crossed). The rare paths are generated such that any trajectory from A to B passes through each

interface in turn. The transitions between interfaces are free to follow any possible path between A and B, including paths crossing previous interfaces several times. This is illustrated in Figure 4.1.

The rate constant $k_{A\to B}$ for transitions from A to B can be calculated from the total average flux from A to B, which can be expressed as the product of a flux from A to λ_0, $\overline{\Phi}_{A,0}$, and the probability that a trajectory reaching λ_0 from A will reach to B without returning to A, $P(\lambda_{n=B} \mid \lambda_0)$,:[14]

$$k_{A\to B} = \frac{\overline{\Phi}_{A,0}}{\overline{h}_A} P(\lambda_{n=B} \mid \lambda_0) \qquad (1)$$

Here, the factor h_A is a history-dependent function such that h_A=1 if the system was more recently in A than in B, and h_A=0 otherwise; $\overline{h}_A$ is the average fraction of the time that the system spends in the basin of attraction of A. $P(\lambda_{n=B} \mid \lambda_0)$ can be expressed as the product of conditional probabilities:

$$P(\lambda_{n=B} \mid \lambda_0) = \prod_{i=0}^{n-1} P(\lambda_{i+1} \mid \lambda_i) \qquad (2)$$

where $P(\lambda_{i+1} \mid \lambda_i)$ is the probability that a trajectory that visits A and crosses λ_i for the first time will subsequently reach λ_{i+1} without returning to the initial region A. [12]

At the present, three path sampling schemes have been proposed to generate transition paths belonging to the TPE and to determine the rate constants. These schemes are: the Forward Flux Sampling (FFS), Branched Growth method (BG), and Rosenbluth method (RB).[13,14] In this work, we used the BG sampling scheme because it has demonstrated to be efficient for simple systems[13] and leads to sampling of the TPE with better resolution at later interfaces in the phase space region, close to the TSE and region B.[13] In the next section (Sec. II A), we briefly describe the BG

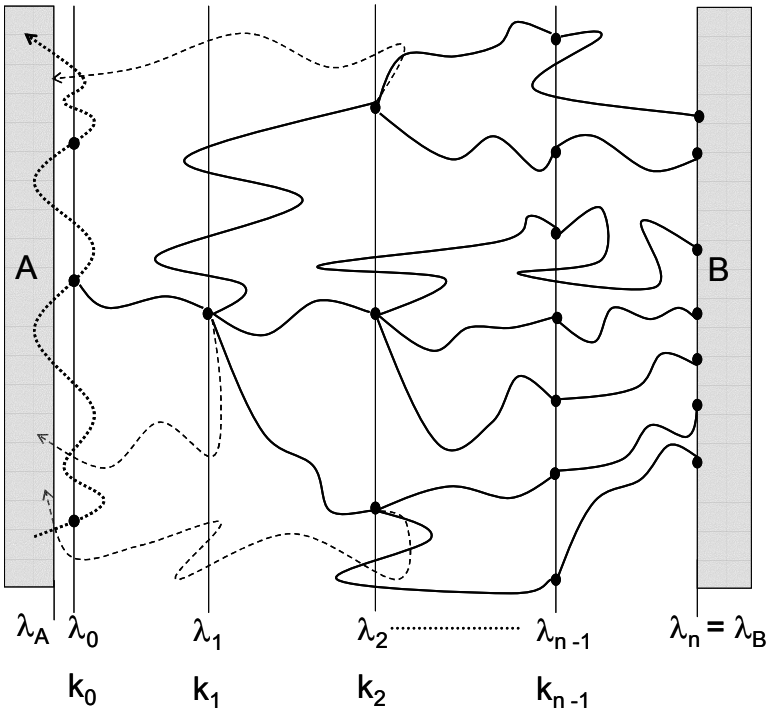

Figure 4.1 A schematic view of the generation of branched paths (thick lines) using the branched growth sampling method (BG). The first stage is the simulation run in the A basin shown by a dotted line. Starting points for the subsequent generation of branched paths are labeled with a black circle at λ_0. The second stage correspond to the trial runs (k_i) fired from λ_i ; those that reached the next λ_{i+1} interface are shown by a thick line and those which failed to reach λ_{i+1} are shown by a dotted line.

sampling approach. For a detailed description of the BG method and the other two schemes (FFS and RB), see reference 13.

B. The Branched Growth Sampling Method

In this section we briefly describe the path sampling and rate constant calculation scheme of the BG approach. The BG method is illustrated schematically in Figure 1, where branched transition paths are generated one by one. In the first stage of the algorithm, a simulation is carried out in the basin of attraction of A. After an equilibration period, the run in the basin A is suspended when the system leaves A and crosses λ_0 and the system configuration at that point is stored. In the second stage of the algorithm, a branched path is generated from this initial configuration by generating k_0 trial runs, which are continued until either reaching λ_1 or returning to the initial region. Then, each end point configuration $N_S^{(0)}$ resulting from successful trial runs to reach λ_1 is stored and used as starting point for k_1 trial runs toward λ_2 (or back to A). If $N_S^{(1)}>0$ of the trial runs reach λ_2, the branching tree path is continued by initiating k_2 trials runs to λ_3 from each of the $N_S^{(1)}$ successful configurations. This procedure is repeated until either the final region $\lambda_n=\lambda_B$ is reached or because no successful trials were generated at some intermediate interfaces. Each trial run at λ_i is weighted by a factor of $1/\prod_{j=0}^{i} k_j$ .Once the generation of one branching path is over, an estimate of $P(\lambda_{n=B} \mid \lambda_0)$ is obtained as the total number of branches that eventually reach λ_n, divided by the total possible number of branches, i.e.,

$$P(\lambda_{n=B} \mid \lambda_0) = \frac{N_S^{(n-1)}}{\prod_{i=0}^{n-1} k_i} \qquad (3)$$

To generate the next branching path, the simulation in the basin of attraction A is restarted and a new starting point at λ_0 is stored. This point is generated by the next

time that the system crosses λ_0, provided that the system returned to A between subsequent starting points [black circles at λ_0 in Figure 4.1]. The same procedure described before is used to create a branching tree from this new point, resulting in a new estimate of $P(\lambda_{n=B} \mid \lambda_0)$. After many branching paths have been generated, a final estimate of $P(\lambda_{n=B} \mid \lambda_0)$ is obtained from the average of all the paths. The flux $\overline{\Phi}_{A,0} / \overline{h}_A$ in Eq. (1) can be obtained from the simulation run in basin A by the total number of crossing configurations at λ_0 divided by the total length of this run.

C. TPE and TSE collection

In order to obtain the TPE from the BG scheme, one begins with the collection of trials which arrive at $\lambda_B=\lambda_n$ from λ_{n-1} and traces back the sequence of connected partial paths which link them to region A. This TPE may be used to characterize reactive dynamics at a macroscopic level by computing rate constants as shown in the above section. However, the TPE could also be analyzed to characterize the mechanism by which the rare event occurs, and thus understand transitions at a microscopic level. Assuming that the pathways harvested by the FFS-type sampling include examples of the transition state (TS) intermediates, a properly weighted ensemble of transition states (TSE) could be collected by only screening configurations along reactive pathways. Following Dellago et al., to determine configuration intermediates along the pathways that belong to the TSE, a function $p_B(x,t_s)$ called the committor for state B is defined.[21] This function is the probability that a system with initial configuration x will reside in state B at time t_s. Once such a trajectory has committed to region A or B, subsequent spontaneous transitions are improbable. Hence, the time that the system spends on the barrier can be long from a computational perspective, but is still negligible compared to the expected time the system needs to enter to the transition state region from one of the stable states. In this

case, p_B is essentially only a function of the configuration x, quantifying the tendency of a configuration to relax to a particular basin of attraction under the system's intrinsic dynamics. The location of the TSE can be then determined by screening typical configurations for p_B=1/2.

The procedure to determine the committor, p_B, is illustrated in Figure 4.2(a). In Eq. (4), $p_B^{(N)}$ is estimated for a particular configuration by initiating a finite number (N) of fleeting trajectories from that configuration. The function $h_B^{(i)}$ has a value of one if the trajectory ends in the basin of region B, otherwise it has a value of zero.

$$p_B^{(N)} \approx \frac{1}{N}\sum_{i=1}^{N} h_B^{(i)} \qquad (4)$$

If N is large enough, p_B may be considered as a Gaussian random variable, with fluctuations given by:

$$\sigma^{(N)} = \sqrt{\left\langle \left(p_B^{(N)} - p_B\right)^2 \right\rangle} = \sqrt{\frac{p_B^{(N)}\left(1 - p_B^{(N)}\right)}{N}} \qquad (5)$$

If $\sigma^{(N)}$ is taken as the desired level of statistical accuracy, we can determine the number of fleeting trajectories required for a particular configuration. Based on this, a simple algorithm to determine TSE on the TPE can be implemented as follows. A starting configuration outside state region A is chosen first. From this configuration, a minimum number of fleeting trajectories, N_{min}, is generated and the committor p_B is estimated from the fraction of paths that end in state B. At this stage, the configuration is excluded as a member of the TSE if the interval of confidence $\left[p_B^{(N)} - \alpha\sigma^{(N)}, p_B^{(N)} + \alpha\sigma^{(N)}\right]$ does not include the value ½ (the constant α sets the

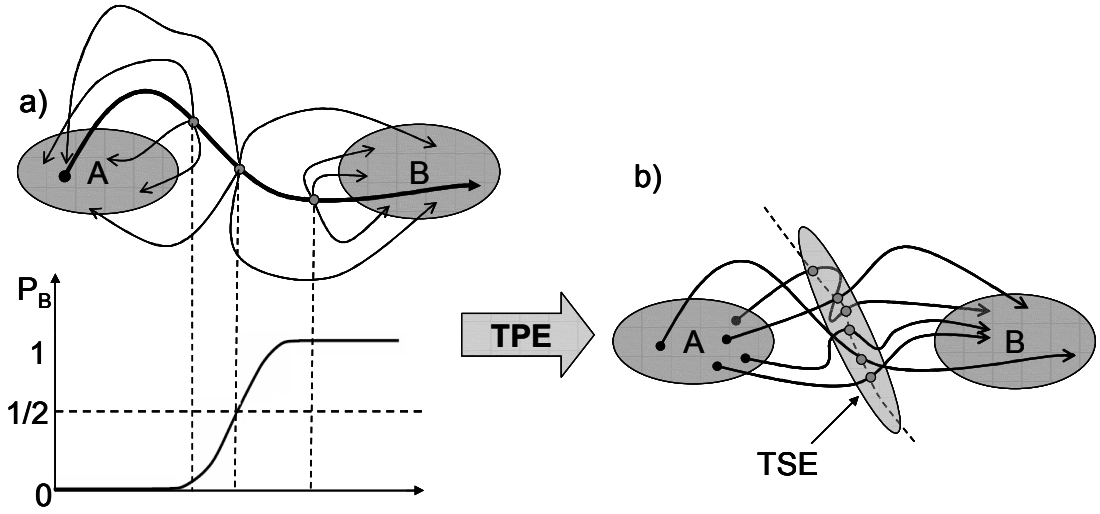

Figure 4.2 A schematic view of the procedure for the determination of committors, p_B, and the TSE. (a) The committor for a stage along a transition path (thick solid line) is estimated from the fraction of fleeting trial trajectories (thin lines) that reach region B. $p_B \approx 0$ for stages close to the basin A, because all trajectories started from that point end in A, $p_B \approx 1$ for stages on the right, because all trajectories started from that point end in B. This sketch is based on figure 18 in reference 21. (b) Illustration of the TSE obtained from the analysis of all paths on the TPE. The configurations for which $p_B \approx 1/2$ are called transition states.

desired level of confidence). Otherwise (if the interval includes the ½ value) further N_{min} fleeting trajectories are subsequently generated from the same configuration and a new estimate for p_B is obtained. This procedure is repeated until either the value ½ falls outside the confidence interval or an upper limit of trial fleeting trajectories, N_{max}, is reached. At this point, if the confidence interval still includes ½, the configuration is accepted as a member of the TSE. Otherwise, it is rejected. The same procedure is repeated for all the states along the path until region B is reached. It is also repeated for all the paths belonging to the TPE, in such a way that a set of transition state configurations from which the system relaxes into one or the other stable state with equal probability is obtained [see Figure 4.2(b)]. In this way, the computational cost of identifying transition state intermediates is reduced because configurations far from the TSE may be excluded quickly. Indeed, configurations lying within the basin A or B require much fewer trajectories than those lying near the TS (i.e., p_B =1/2).[21]

III. Models and Methods

To study folding kinetic over relatively long times and compare our results with other studies of the effect of confinement on folding, we adopted a simple cubic lattice protein model and a Monte Carlo (MC) simulation method.[27] Each amino acid is represented by a bead and the bonds between them have a uniform length equal to the lattice spacing σ. The vacant lattice sites are assumed to be occupied by water, whose net effect is considered by assigning hydrophobic and hydrophilic characteristics to different amino acids. The Miyazawa-Jerningan (MJ) contact energy potential was used to account implicitly for solvent effects[28] with the renormalization of the solvent-solute interactions introduced by Leonhard et al. [29] that corrects for some of the inconsistencies resulting from the approximations underlying the original MJ model.

A. Protein sequence

We used in this study a model 48-mer protein with the following sequence: TSKRQQPYPMSLGSPFIRIPMIGPRPRMRLLILLMGYPKRGRSGGGLF.We chose this sequence because of its stable and fast-folding design[30] and for its well characterized thermodynamic behavior. It was designed to exhibits a single-domain native state and a maximum number of native contacts (NNC) of 57, which correspond to a minimum global energy of U_{nat} = -20.24k_BT. In addition, this model protein was designed to follow an all-or-none transition between two clearly distinguishable states: the native and unfolded states. As a result, this designed sequence exhibits protein-like thermodynamic behavior (i.e., one single peak in the heat capacity temperature dependence), which correspond to a two-state (unfolded-folded) cooperatively interplay between local conformation preferences and non-local interactions.[31] These characteristics made this model protein an ideal benchmark for kinetic studies of folding via a single transition state. Folded structures and a detailed thermodynamic characterization for this model protein sequence can be found elsewhere.[12,30]

B. Simulation details

All simulations were carried out in the canonical ensemble. Conformational sampling was performed through a set of MC moves based on the Verdier-Stockmayer algorithm that mimic the diffusive movement of the amino acids during the folding process and includes: (i) tail moves of one of the end beads to one of the available four neighboring sites, (ii) corner flips for beads characterized by a right angle between directions to both contour neighbors, and (iii) crankshaft moves of bead pairs located at the bottom of a U turn.[16]

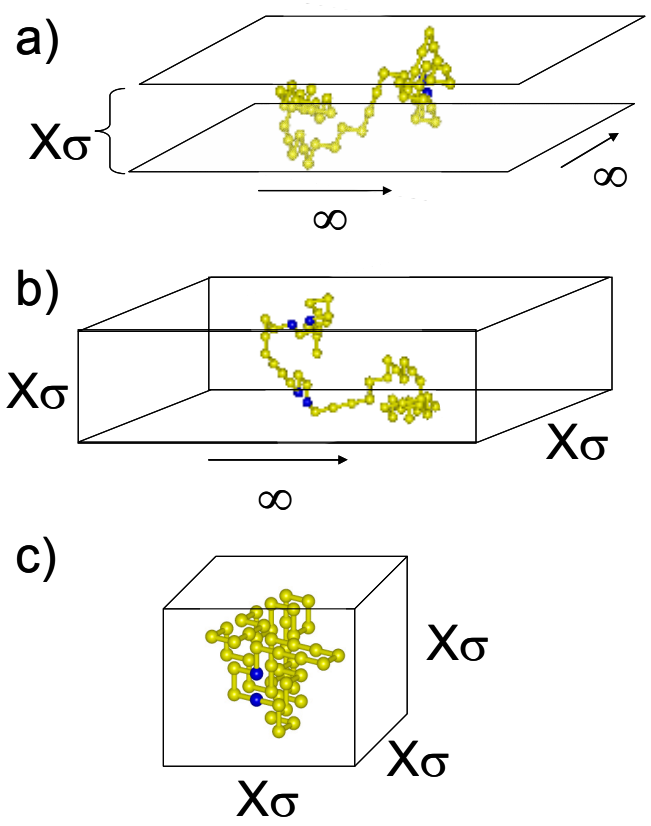

Figure 4.3 Confinement geometries: (a) between two infinite planes (1-D confinement); (b) inside an infinite tunnel (2-D confinement); and (c) inside a cubic cage (3-D confinement). $X\sigma$ is the distance between inert walls. The initial unfolded configuration from which the FFS-simulations were started for each of the confinement geometries is also shown (native contact pairs shown in dark/blue). These configurations were chosen randomly from an ensemble of unfolded conformations with Q<0.05 generated at high temperatures.

The effect of surface-protein interactions on the folding pathway was studied by confining the protein inside a space restricted by non-interactive (i.e., inert) walls. Because the focus of these simulations is to explore the feasibility of FFS-type schemes to elucidate the role of surface interactions on the protein folding process, simple confinement geometries were assumed. The confinement was enacted in different dimensions (D) to mimic the space restriction that a protein could experience inside of the ribosomal tunnel, a chaperon cavity, or in the crowded cell's environment.[3,6-8,10,12,32,33] Figure 4.3 illustrates the confinement geometries assumed: (i) between two infinite planes (1-D confinement); (ii) inside an infinite square tunnel (2-D confinement); and (iii) inside a cubic cage (3-D confinement). In Figure 3, $X\sigma$ stands for the distance between walls that ranged from 6σ to infinite (i.e., open space). The initial unfolded configuration from which the FFS-simulations were started is also show in Figure 4.3.

C. Thermodynamic analysis

Independent from the BG simulations (that only used the Verdier-Stockmayer pseudo-dynamic moves), we studied the thermodynamics of the model protein by employing Replica Exchange Monte Carlo (REMC) sampling [16,34] combined with the Multi-Histogram Reweighting method (MHR).[35] REMC was used to alleviate problems related to the sampling of a rugged free-energy landscape, in which the polypeptide chain could be temporarily trapped at low temperature. Protein folding was simulated by running several (M) parallel replicas, each one at a different temperature, T_i. The reduced temperature T is normalized by the reference temperature T_o such that $k_B T_o$ represents the energy unit pertinent to the system. Relative to Verdier-Stockmayer moves, swaps moves between systems of different Ts were attempted after each MC step with a probability ≤ 0.05. In most calculations, the number of replicas was 9,

with T ranging between 0.20 and 0.5. The T levels in this range were determined with an adaptive algorithm that tracks the diffusion of replicas in temperature space and iteratively improves the efficiency of parallel tempering by concentrating temperature points in the regions of diffusional bottlenecks.[36] The resulting T levels were: (i) {0.200, 0.254, 0.257, 0.260, 0.266, 0.267, 0.272, 0.286, 0.500} for the bulk; (ii) {0.200, 0.252, 0.256, 0.260, 0.263, 0.266, 0.271, 0.284, 0.500} for the 10σ 3-D cage; and (iii) {0.200, 0.251, 0.257, 0.262, 0.268, 0.276, 0.291, 0.312, 0.500} for the 6σ 3-D cage.

Various thermodynamic quantities can be obtained for different temperatures, based on a single REMC simulation run. By using the MHR method, data from all replicas can be combined and analyzed, minimizing the error in the estimation of the density of state function Ω(U) and facilitating the calculation of free-energy surfaces. Once Ω(U) is known, averages of other quantities X, such as order parameters can be determined from:

$$\langle X \rangle_T = \frac{\sum X(U)\Omega(U)e^{-\beta U}}{\sum \Omega(U)e^{-\beta U}} \qquad (6)$$

In this work, the radius of gyration, R_g, and the fraction of native contacts present, $Q = NC / NNC$, were calculated using Eq. (6) and used to analyze the structure of the protein. Ω(U) was also used to calculate the specific heat (C_v) via Eq. (7), and free energy (A_E) via Eq. (8), over a wide range of temperature in the form of continuous functions.

$$C_V(T) = \frac{\langle U^2 \rangle_T - \langle U \rangle_T^2}{k_B T^2} \qquad (7)$$

$$A_E(U,T) = U - TS = -k_B T\left(\ln\left(P(U,T) - \ln(Z(T)\right)\right) \qquad (8)$$

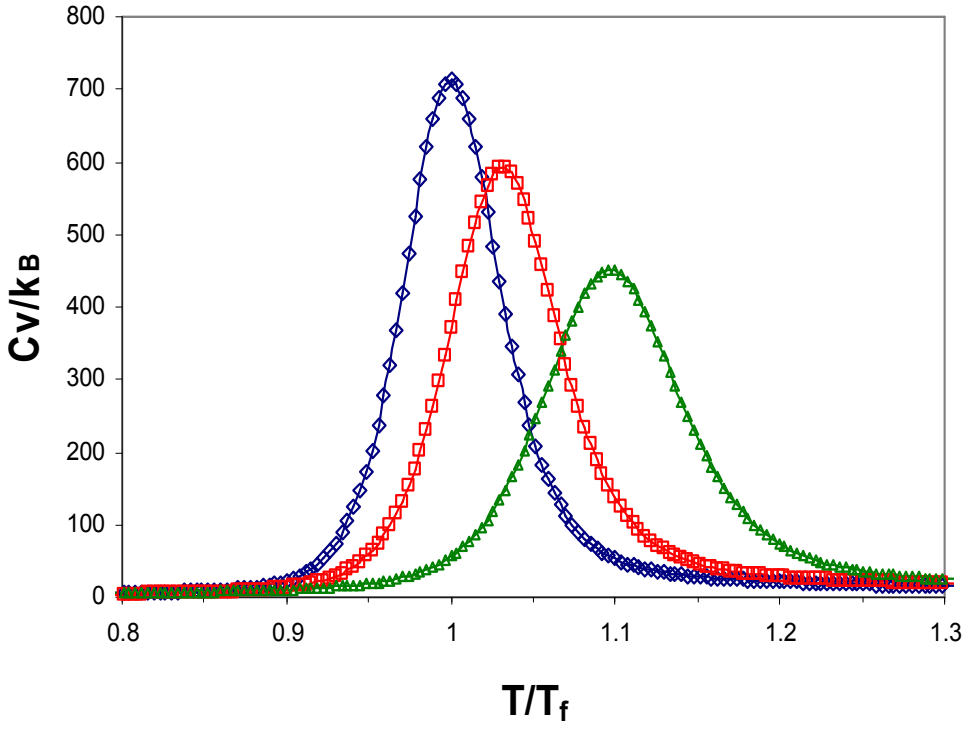

Figure 4.4 Heat capacity as function of temperature for: (◊) bulk, and (□) 10σ and (Δ) 6σ 3-D cage. Tf is the temperature where Cv peaks signaling the change between a random coil and the folded state.

Here, $S(U)$ is the entropy, $Z(T)$ is the partition function $=\Sigma_U\ \Omega(U)\exp(-U/k_BT)$ and $P(U,T)$ is the Boltzmann distribution of states $= \Omega(U)\exp(-U/k_bT)/\ Z(T)$.

IV. Simulation results

A. Free-energy calculations

For concreteness, the effect of confinement on the thermodynamic properties is presented for the 3-D confinement case only. We estimated the transition temperature by T_{max}, the maximum in the C_v vs. T plot. Figure 4.4 shows a single and strong peak for the protein in the bulk and when confined in cubic boxes of lengths 6σ and 10σ lattice units, suggesting a single transition from a random coil to the folded state. As the confinement volume decreases, the peak's maximum is shifted to higher temperatures thus causing an increase in the temperature range of stability of the folded state. We define the *bulk transition temperature* T_f as T_{max} for the protein in open space and thus T_f =0.27. The peak then occurs at T_{max}/T_f = 1.03 for the 10σ 3-D cage, and at T_{max}/T_f = 1.10 for a 6σ 3-D cage. Figure 4.5 shows free energy landscapes for the peptide chain simulated in isolation and under restricted space. In these plots, the progress variables are the native energy and the fractional nativeness, Q. The enhancement in thermodynamic stability under confinement is related to an increase in the frequency with which the protein has configurational energies close to that of the native state and to a decrease in the frequency with which the denaturated protein explores extended configurations (Q~0.1) during the folding dynamic, as seen in Figure 4.5. Note that the position of the transition state (TS) in these plots was determined by the procedure described in the Sec. IV C. The stabilization effect of confinement is also reflected in a decrease in free energy of the native state (F) relative to the free energy of the denaturated state (U) as observed in Figure 4.6 (i.e., increase

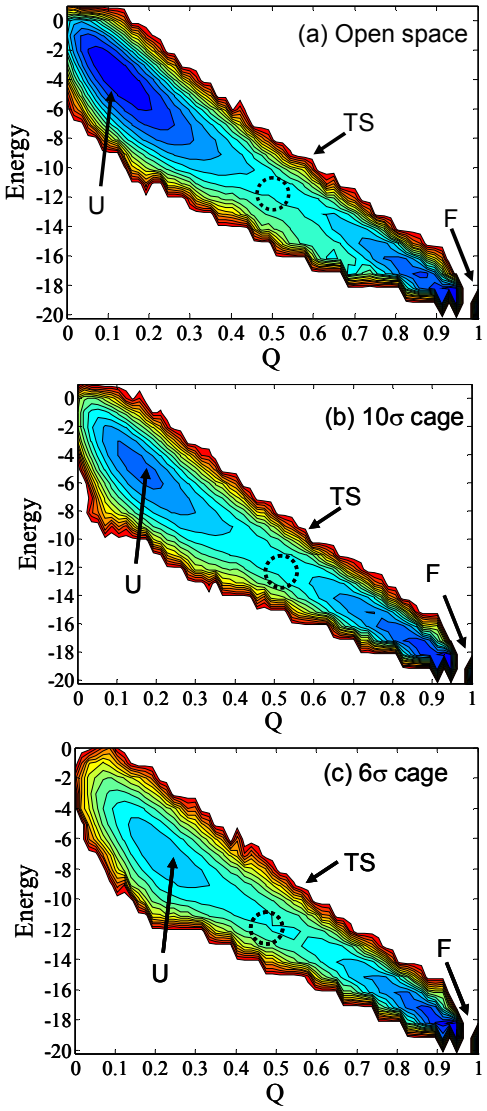

Figure 4.5 Contour plot of the free energy landscape [energy vs. fractional nativeness (Q)] at T_f = 0.27 for: (a) open space, and (b) 10σ and (c) 6σ 3-D cage; the color scheme changes from highest (gray/red) to lowest (black/blue) elevations. The regions corresponding to folded state (F), unfolded state (U) and transition state (TS) are also shown.

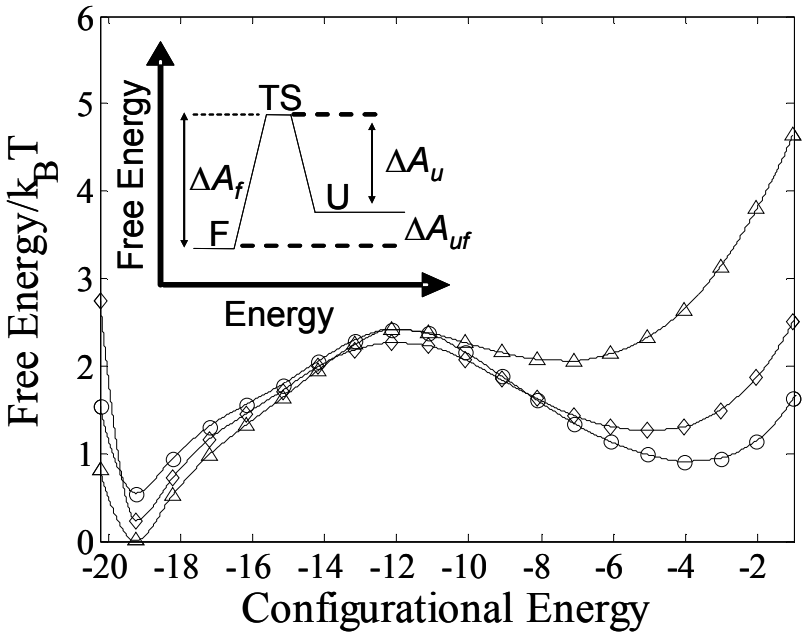

Figure 4.6 Free energy versus configurational energy a T_f = 0.27 for: (o) bulk, and (◊) 10σ and (Δ) 6σ 3-D cage. The inset gives a schematic diagram of the free energy of the native (ΔA_f), and unfolded state (ΔA_u), and the free energy of stabilization (ΔA_{uf}). F=Folded state, U= Unfolded state, and TS= Transition state.

of ΔA_{uf}). Moreover, it is observed that the change in free energy, ΔA_u, between the transition state (TS) and the unfolded state at T_f decreases as the confinement volume decreases. This is due to the reduction of the number of available non-native configurations, causing a displacement of the unfolded state towards the transition state (see Figure 4.5) which leads to a decrease in the system entropy.

The free energy of stabilization (ΔA_{uf}) at T_f, as well as the enthalpic (ΔU) and entropic (ΔS) contribution are summarized in Table I. ΔA_{uf} at any temperature can be computed from

$$\Delta A_{uf} = A_{folded} - A_{unfolded} = -k_B T \ln(\frac{P_f}{1-P_f}) \qquad (9)$$

where P_f is the probability of the folded state at temperature T. In our analysis, the threshold value for defining a folded state was chosen as the fractional nativeness at the folding temperature, $\langle Q(T_f)\rangle \approx 0.5460$. Hence, a protein is considered folded if $Q > Q(T_f)$. This treatment yields $\Delta A_{uf} = 0$ at the transition temperature for a protein folding in open space and facilitates comparison of the confinement effects. The enthalpy change associated with the folding can be computed from the difference between the average configurational energy of the folded and unfolded states. The entropic contribution to ΔA_{uf} can be estimated from $T\Delta S = \Delta U - \Delta A_{uf}$. It is also shown in Table 4.1 that for each confined case, the entropic cost of folding is less than that of its bulk counterpart at $T_f = 0.27$. This occurs because the conformational space limits the space available to the unfolded state and hence destabilizes the unfolded state by reducing its entropy. However, the enthalpy of folding is not as favorable for any degree of confinement as it is in the bulk because the energy of the unfolded state decreases with confinement. Thus, the enhanced thermodynamic stability of the native

Table 4.1 Changes in free energy, enthalpy and entropy for protein confined to a 3-D cage at Tf = 0.27.

	ΔA_{uf}	ΔU	$T\Delta S$
Bulk	0.00	-14.10	-14.10
10σ	-0.36	-13.12	-12.76
6σ	-0.97	-10.62	-9.65

state on confinement arises from two competing factors. Folding is favored entropically but hindered energetically, and since the entropic contribution is larger in magnitude, the result is an overall increased stability for the confined folded state. A more detailed, quantitative analysis of the restriction to extended conformations under confinement could be seen in a plot of the population density as a function of *Rg* and *Q* (not shown). In the bulk at T_f, ~ 50% of the population is inside the native-like basin (Q ≥ 0.9; 3.0 Å < R_g < 3.6 Å) and the rest of the configurations correspond to large- R_g unfolded states. Upon confinement, the unfolded state losses entropy and the population within the native-like basin increases to 61% and 81% for the 10σ and 6σ 3-D cage at T_f, respectively.

To summarize, the confinement at T_f restricts the number of extended configurations available to the denaturated protein (i.e., smaller values of R_g for the unfolded state) causing a reduction of the configurational entropy of the system (i.e. smaller ΔA_u). Similar results were obtained for the 1-D and 2-D confinement cases (not shown), where the thermodynamic stability of the folded state was seen to increase with the number of constrained dimensions.

B. Folding kinetics under restricted conformational space

The folding kinetics of the system was simulated with the BG sampling scheme. The fraction of nativeness was used as order parameter (i.e., λ=Q), and the unfolded state (λ_A) was defined depending on the confinement conditions. It is observed in Figure 4.5 that the center's position of the unfolded basin of attraction rests close to Q~ 0.1, 0.15, and 0.20 for the bulk, and the 10σ and 6σ 3-D cages, respectively. Accordingly, we defined $\lambda_A=\lambda_0$=0.1 for the protein folding dynamics in open space and $\lambda_A=\lambda_0$=0.15 and 0.20 for the 10σ and 6σ 3-D confinement, respectively. The folded state was defined by taking $\lambda_B=\lambda_n$=1.0. We used nine

interfaces to partition the phase space (n=9), with interfaces positioned at λ_i ($0 \le \lambda_i \le$ n): (i) {0.10, 0.13, 0.15, 0.2, 0.3, 0.4, 0.5, 0.65, 0.8, 1.0} for the bulk; (ii) {0.15, 0.18, 0.20, 0.25, 0.3, 0.4, 0.5, 0.65, 0.8, 1.0} for 10σ 3-D cage; and (iii) {0.2, 0.23, 0.25, 0.3, 0.35, 0.4, 0.5, 0.65, 0.8, 1.0} for 6σ 3-D cage. In the case where the space was restricted in 1-D and 2-D, the interfaces were positioned as in the bulk case. The number of trials per point was k_i = {5, 3, 2, 2, 3, 3, 2, 2, 1} ($0 \le i < n$). In all cases, the rate constant (k_{UF}) calculations were carried out as a series of blocks, each consisting of 1000 starting points at λ_0; final results were averaged over all blocks. Following Allen et al.,[13] the error bar for the estimate of k_{UF} is given by $k_{UF}\sqrt{\nu / N_0}$. The statistical error ν is defined to be the variance $V(k_{UF})$ in the estimate of $k_{UF,}$ per initial point at $\lambda_{0,}$ divided by the square of the expectation value $E(k_{UF})$:

$$\nu \equiv N_0 \frac{V[k_{UF}]}{(E(k_{UF}))^2} \approx N_0 \frac{\Phi^2 V[P(\lambda_n \mid \lambda_0]}{(\Phi E(P(\lambda_n \mid \lambda_0))^2} = N_0 \frac{V[P(\lambda_n \mid \lambda_0)]}{P(\lambda_n \mid \lambda_0)^2} \qquad (10)$$

where N_0 is the number of starting points at λ_0 and $V[P(\lambda_n \mid \lambda_0)] = \overline{(P(\lambda_n \mid \lambda_0))^2} - \overline{(P(\lambda_n \mid \lambda_0)}^2$ is the variance in the estimated value of $P(\lambda_{n=B} \mid \lambda_0)$.

The rate constant of transitions between unfolded and folded states was determined over the range of temperatures where the native state is stable (i.e. $T<T_f$). Figure 4.7 shows the result for the folding rate constant under several types of confinement with a characteristic width between inert walls of 6σ lattice spaces. Figure 4.7 also illustrates the results for the folding kinetics under 3-D confinement with widths: 10σ and 6σ. For comparison the rate constant k_{UF} was also determined for the protein folding in open space over the same range of temperature. The results from these simulations show that the confinement of a protein inside a cage decreases the folding kinetics for temperatures below the minimal folding-time temperature (i.e.,

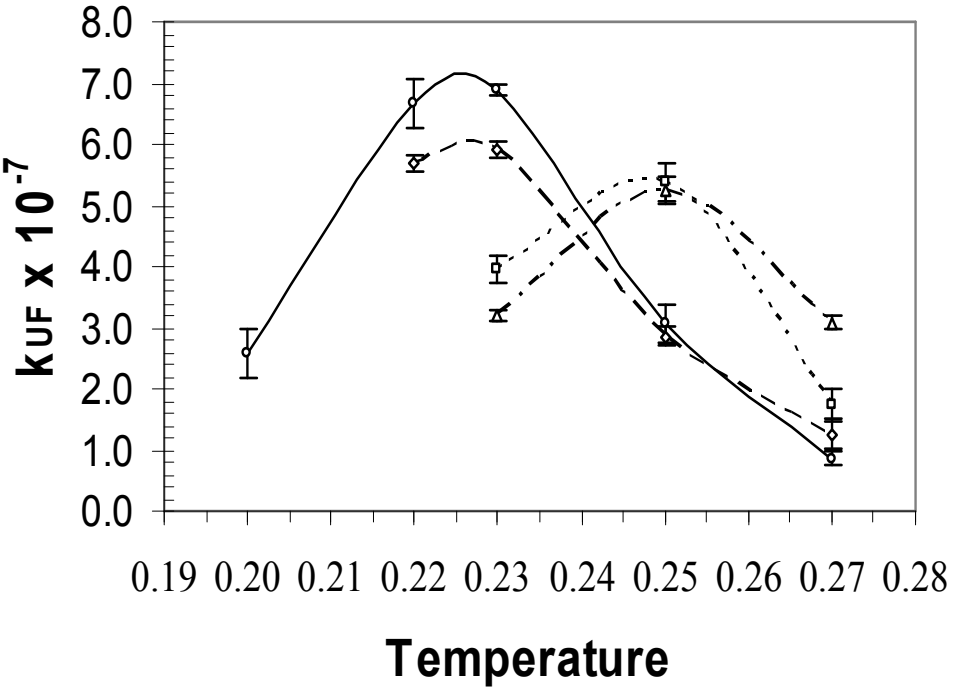

Figure 4.7 Folding rate constants from the BG method over a range of temperature and restricted spaces with confinement length: (— ● – –) infinity (bulk), (♦·-·-) infinite planes (6σ 1-D), (–■–) infinite tunnel (6σ 2-D), (- -▲- -) 10σ 3-D cage, and (▲—) 6σ 3-D cage. Brute force estimates for selected points are shown by asterisks (∗).

temperature at which the protein fold fastest in an open space, T_{opt} ~ 0.23). On the other hand, relative to the free-space case, the kinetics can be accelerated at temperatures above $T_{opt,}$ where k_{UF} increases with the number of constrained dimensions, as seen Figure7. However, for a temperature in this range (T_f>T>T_{opt}), there exists a cage width (case of 3-D confinement), at which k_{UF} is maximized; e.g., for T=0.24 in Figure 4.7, the maximum folding rate was obtained for a 3-D confinement space of 10σ lattice units. Hence, strong confinement is detrimental to folding kinetics because the protein could be trapped in local minimum energy conformations, where the global reconfiguration needed to attain its ground state is not readily possible. In this case, the protein becomes increasingly trapped in low energy minima, requiring high temperatures to undergo large conformational fluctuations and make transitions to the native state. Our results are entirely consistent with previously reported works, in which the kinetics of the system was quantified via median folding times.[7,37] For instance, early chaperonin cage studies by Baumketner et al. showed that confinement served to accelerate folding for temperatures exceeding T_{opt} when a protein is confined in a volume above a certain threshold and decelerated below.

For two-state proteins as the one considered here, there is just one rate-determining transition state to be considered between the native and the denatured states. It can then be assumed that the system will follow single-step first-order kinetics for folding and unfolding. Consequently, the mean first passage time (MFT), i.e., the average time when a system will first reach the final stage, given that it is in an initial state at t=0, can be obtained from:

$$MFT = \int_{t=0}^{\infty} k_{UF} t e^{-k_{UF} t} dt = \frac{1}{k_{UF}} \qquad (11)$$

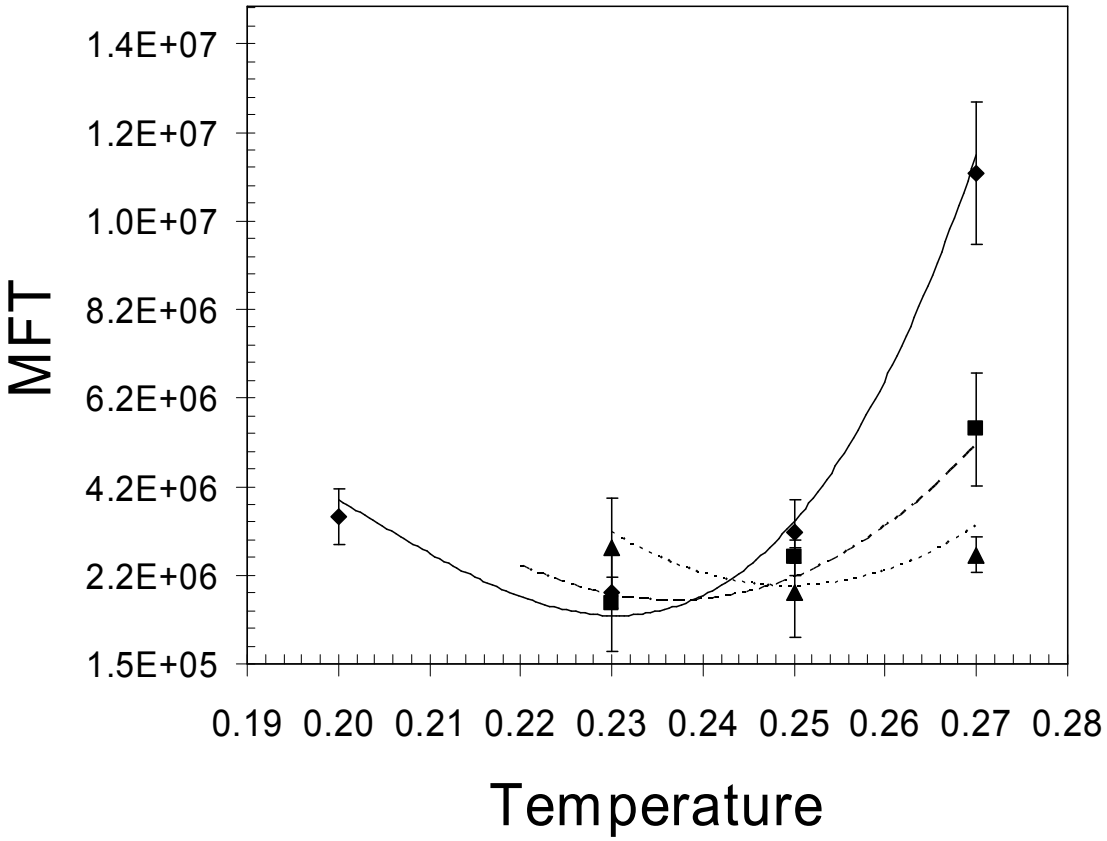

Figure 4.8 Mean folding time (MFT) over a range of temperature for 3-D cages with widths: (♦) infinity (bulk), (■) 10σ , and (▲) 6σ (estimated from over 500 runs). Estimates for the mean folding time from the inverse of the transition rate constant are also shown: (—) bulk, (---) 10σ , and (····) 6σ .

One could also find the MFT (in terms of number of MC steps) by directly calculating the average folding time when each simulation first reached the folded state. Figure 4.8 compares the mean folding time calculated from 500 independently runs from the inverse of the transition rate constant for the folding kinetics in free space and under 3-D confinement with widths: 10σ and 6σ. This figure evidences that the model protein follows a two-state transition given by a single transition state (this is further supported in the following section). The assumption of first-order kinetics could lead to erroneous results, however, in systems that follow folding mechanism different from a two-state transition (i.e., folding pathways through intermediates or multiple TS). In those cases, the characterization of the system's kinetics should be preferably done by the determination of k_{UF} rather than the MFT.

The calculations of k_{UF} using the brute-force approach entailed the generation of the folding time distribution over a total of 4.8×10^7 simulated folding events, each one of them starting from a different unfolded structure (i.e NNC≤2). The integral $F(t) = \int_0^t p(t')dt'$ of $p(\tau)$ distribution was fitted to the Poisson function $F(t) = 1 - \exp[-k_{UF}t]$, in order to measure k_{UF}. The rate constant k_{UF} values obtained from brute force calculations are also presented in Figure 7 by the symbol (*). It is clear that all the results from the path sampling method are in excellent agreement with the brute-force results. Following Allen et al.,[13] the computational efficiency (ε) was defined as $1/[C\nu]$ where C represents the computational cost, which is estimate to be the average number of simulation steps (i.e. MC steps), per initial point at λ_0. The statistical error ν is given by the Eq. (10). The computational cost for brute-force calculations was taken as the average folding time over the total numbers of simulated folding events (4.8×10^7). Table 4.2 compares the k_{UF} values, computational cost, and efficiency obtained from brute-force, and FFS-type and TPS scheme for the protein folding in bulk at $T_{opt} = 0.23$. It is clear that the two path

Table 4.2. Rate constant k_{UF} values obtained from brute-force, and FFS-type and TPS scheme for the protein folding in bulk at T_{opt} = 0.23.

	k_{UF}	$V(k_{UF})$	C	ν	ε	error bar
Brute Force	6.848×10^{-7}	1.409×10^{-14}	1.6×10^{6}	1.442×10^{6}	4.334×10^{-13}	1.187×10^{-7}
TPS	6.866×10^{-7}	1.014×10^{-14}	4.2×10^{4}	2.051×10^{1}	1.161×10^{-6}	1.007×10^{-7}
FFS-type (BG)	6.894×10^{-7}	1.166×10^{-16}	3.5×10^{5}	2.450×10^{-1}	1.166×10^{-5}	0.108×10^{-7}

sampling algorithms are much more efficient than brute-force simulation, even for this relatively simple lattice protein model. However, the FFS-type scheme used in this work provides a more efficient way to obtain rate constant values, resulting in a higher computational efficiency. In appendix A, we briefly describe the TPS scheme used to calculate k_{UF}.

C. Folding mechanism

The TSE ensemble is anticipated to be structurally diverse, reflecting the many ways a transition for a rare event can occur. So, the configurations belonging to the TSE can be examined to determine the distribution of several order parameters, searching for patterns within the TSE. Note that some of the trajectories on the TPE may even cross the TSE more than once. In this case, each one of these intermediates provides a valid example of the TSE.

The branching TPE obtained from the BG path sampling algorithm was inspected to obtain the TSE. A total of 100 branched paths were analyzed, in which configurations along the partial paths between interfaces were stored at a time interval of $3x10^3$ MC steps. The member states of the TSE were accepted by calculating the committor for region B, as described in Sec. II C. In these simulations a confidence level of 95% (i.e., α =2) was used. The minimum and maximum number of fleeting trajectories was fixed to 100 and 1000, respectively. The collection of configurations was then examined by searching the pattern of the fractional nativeness (Q) within TSE. Figure 4.9 reports the distribution of Q within TSE for the protein folding kinetics in free space and under 3-D confinement with widths: 10σ and 6σ at T_f = 0.27. Note that the Q distribution is sharply peaked at 28/57≈0.49, 29/57≈0.51, and 25/57≈0.44 for bulk, and 10σ and 6σ 3-D cage, respectively. These histograms help identify the intermediate regime of rapidly inter-converting conformations that are

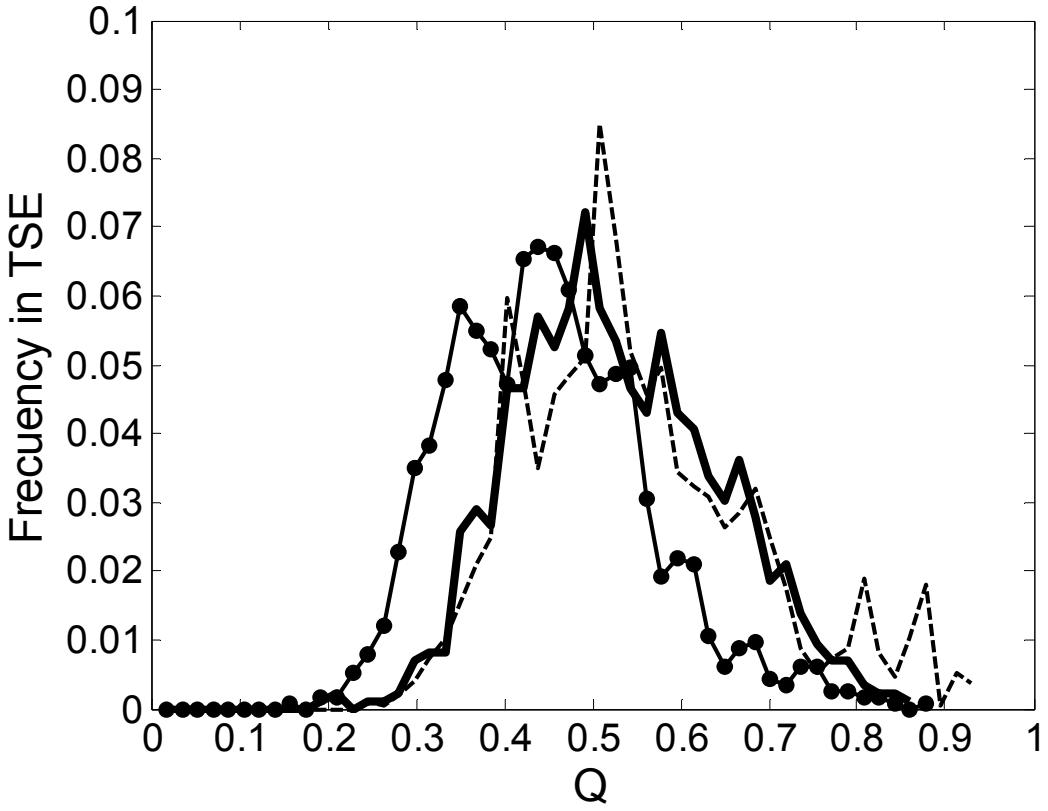

Figure 4.9 Distribution of fractional nativeness (Q) within the transition state ensemble at $T_f = 0.27$ for: (—) bulk, (---) 10σ 3-D cage, and (—•—) 6σ 3-D cage.

transient, and do not accumulate (i.e., sparsely populated). Furthermore, the presence of these kinetic intermediates is consistent with a cooperative equilibrium two-state transition in the sense that only the unfolded (U) and folded (F) states are ever noticeably populated in equilibrium, as seen in Figure 4.5. It is also shown in Figure 4.5 the average position of the transition state (i.e. $\langle Q \rangle_{TSE}$) determined from the Q distribution within TSE. The $\langle Q \rangle_{TSE}$ corresponds to an average conformational energy $\langle U \rangle_{TSE} = -12.14 k_B T$ for all the three cases. This value matches well the value observed at the top of the free energy barrier in Figure 4.6. Hence, Q could be considered as an appropriate reaction coordinate that describes properly the dynamical bottleneck between the two stable states for this lattice protein.

To obtain a more complete picture of the mechanistic details of the folding event, we investigated the specific amino acid residues and residues native contacts (NC) that contribute most significantly to the transition state configurational energy. The 15 amino acid pairs listed in Table 4.3 are the most probable native contacts that on average have a 60% chances or more to occur in the TSE. They are listed in order of decreasing probability. For a protein under 10σ 3-D confinement, 14 of the 15 most dominant native contact pairs are also found in the bulk transition state, which indicates that the confinement of the protein in a moderate sized cavity does not have a significant effect on the folding intermediates from which the protein obtains its native structure. This is in accord to the result that the $\langle Q \rangle_{TSE}$ for both bulk and 3-D confinement with 10σ lattice units of separation between walls is approximated equal to 0.50, corresponding to $\langle U \rangle_{TSE} = -12.14 k_B T$. However, for a protein folding in a 6σ 3-D cage, only 11 of the 15 dominant native contact pairs are also found in the bulk transition state. Figure 4.10 shows the distribution of native contact pairs' probability within the TSE for bulk and 6σ 3-D cage systems. In this case, the distribution the native contact pairs within the TSE is affected, causing a transition intermediate with

Table 4.3 Most probable native contacts found in the transition state ensemble for protein suggested to a 3-D cage at Tf = 0.27. They are listed in order of decreasing probability.

bulk		**10σ**		**6σ**	
NC pairs (i, j)		**NC pairs (i, j)**		**NC pairs (i, j)**	
i (type)	**j (type)**	**i (type)**	**j (type)**	**i (type)**	**j (type)**
20(P)	35(M)	20(P)	35(M)	19(I)	34(L)
21(M)	24(P)	21(M)	24(P)	21(M)	24(P)
19(I)	34(L)	19(I)	34(L)	20(P)	35(M)
20(P)	37(Y)	22(I)	35(M)	17(I)	34(L)
19(I)	30(L)	19(I)	30(L)	19(I)	30(L)
22(I)	35(M)	21(M)	26(P)	20(P)	27(R)
23(G)	36(G)	20(P)	37(Y)	5(Q)	18(R)
20(P)	27(R)	23(G)	36(G)	13(G)	16(F)
21(M)	30(L)	20(P)	27(R)	11(S)	14(S)
21(M)	26(P)	19(I)	28(M)	16(F)	35(M)
19(I)	28(M)	24(P)	37(Y)	21(M)	26(P)
13(G)	16(F)	26(P)	29(R)	22(I)	35(M)
24(P)	37(Y)	17(I)	34(L)	23(G)	36(G)
17(I)	34(L)	37(Y)	40(R)	37(Y)	40(R)
26(P)	29(R)	13(G)	16(F)	20(P)	37(Y)

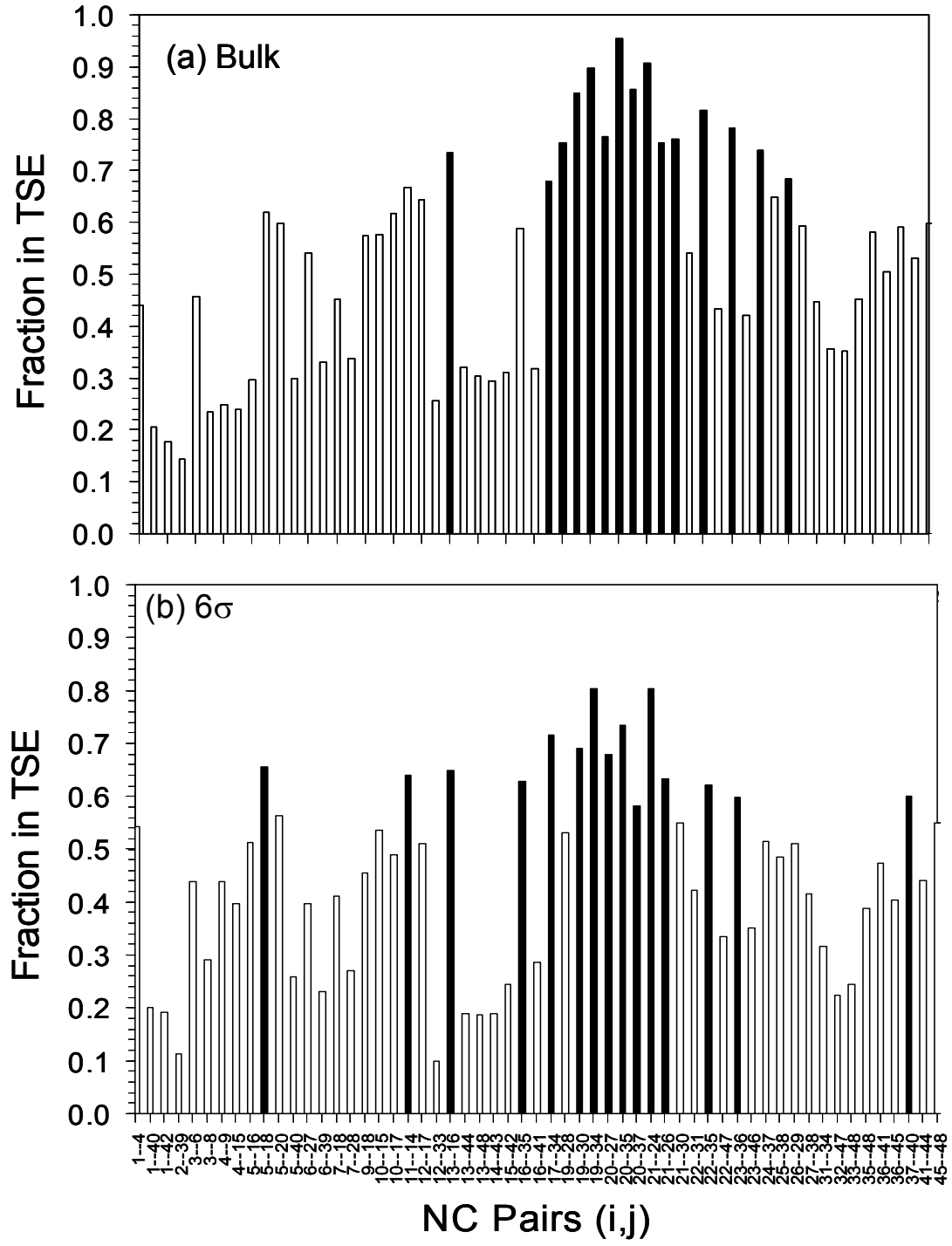

Figure 4.10 Distribution of native contact pairs' frequency within transition state ensemble at T_f = 0.27 for: (a) open space, and (b) 6σ 3-D cage. The black bar represents the 15 native pairs listed in Table 4.3.

an $\langle Q \rangle_{TSE} \approx 0.44$, corresponding to $\langle U \rangle_{TSE} = -12.14 k_B T$. Therefore, the confinement in a 3-D cage with 6σ lattice units of separation between walls at $T_f = 0.27$, restricts the number of extended configurations available to the denaturated protein causing a displacement of both the average unfolded state and the transition state close to it. However, there are some native contacts pairs that are essential for the transition from unfolded to native state. For instance, the first three NC pairs between hydrophobic residues listed in Table 4.3 [i.e., (20, 35), (21, 24), and (19, 34)] are observed with the highest probabilities in the TSE for all three cases (hydrophobic residues include G, L, I, M, P, F). Additionally, the pairs formed between amino acids 17 to 37 form a core of native contacts, which evidence a nucleation event during folding.[38,39] Interestingly, 80% of the NC pairs formed between amino acids: 17 to 37 correspond to interactions between hydrophobic residues that (on average) contribute 20% of the potential energy for the folded structure.

Figure 4.11(a) shows snapshots of a folding mechanism for this protein in open space; it can be observed that the 15 NC pairs with greater occurrence probability in the TSE correspond to the nucleus that begins to form at early stages (Q~0.2) during the folding process (the residues are shaded to display the formation of NCs). Figure 4.11(b) also shows the formation of a core at early stages (Q~0.3) during the folding process under 6σ 3-D cage confinement. However, the unfolded intermediates exhibit more compact structures because the long-lived extended intermediates present in the bulk are either suppressed or their lifetimes shortened. This results in increased thermodynamic stability of the folding intermediates, where the protein could be stabilized against reversible folding by folding forces different from those encountered by proteins in a free space, leading to stable TS with a smaller $\langle Q \rangle_{TSE}$. Note in Figures 4.11(a) and (b) that the darker (red) residues are clustered in a patch,

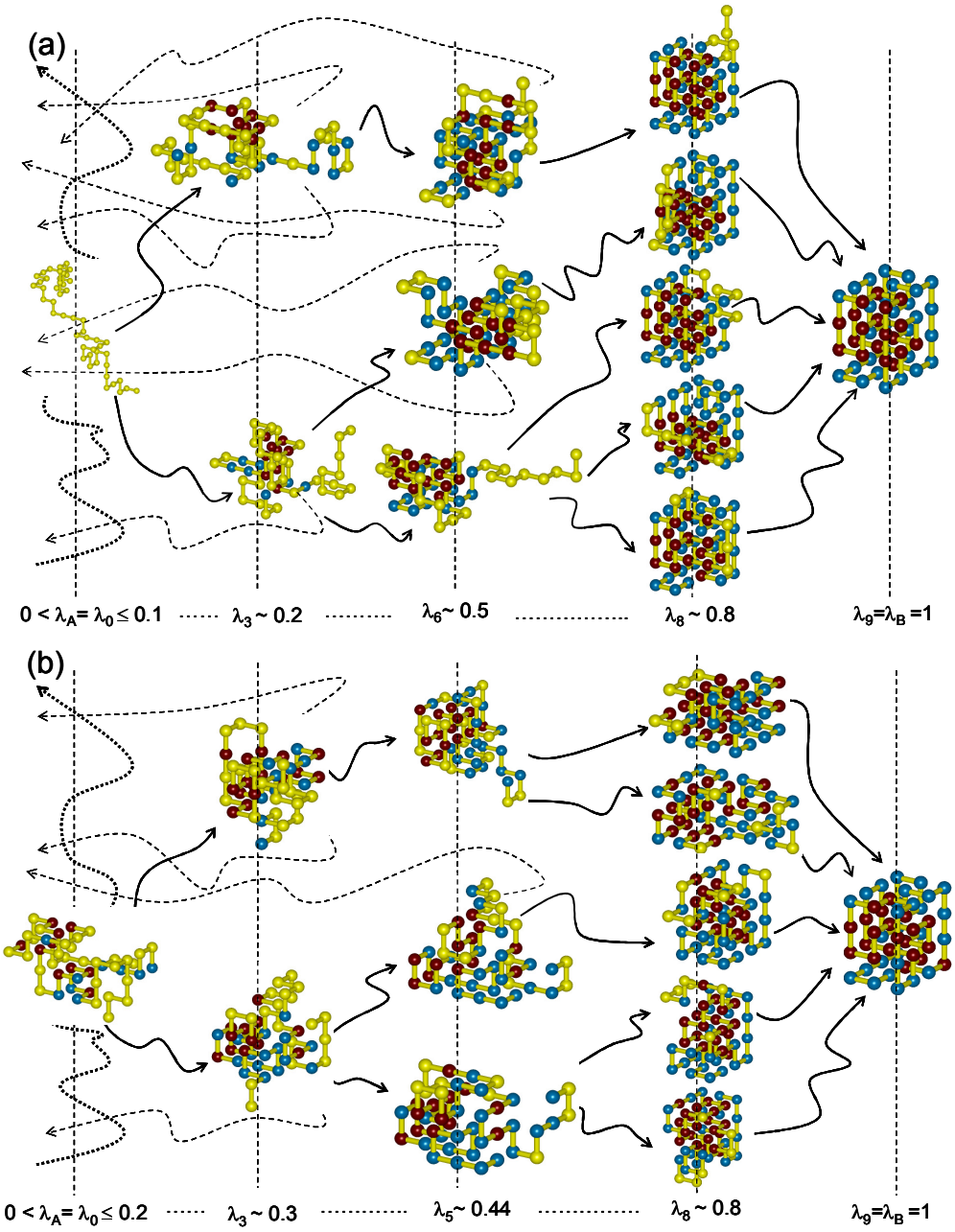

Figure 4.11 Snapshots of a folding event at $T_f = 0.27$ in (a) open space and (b) a 6σ 3-D cage. The simulation run in the unfolded basin is shown by a thick dotted line. Trial runs (k_i) fired from λ_i that reach the next λ_{i+1} interface are shown by a thick line and those which failed to reach λ_{i+1} are shown by a dotted line. Amino acids contacts are encoded by the color of the residue: light gray (yellow) have no native contacts, medium gray (blue) indicates native contacts, whereas dark gray (red) identifies the native contacts listed in Table III, which have the highest probability to belong to the transition sate. The rightmost frame shows the native state conformation.

representing a nucleation event of NCs during the folding process. In both free space and under 6σ 3-D cage confinement then, the peptide chain begins in an initial, highly fluctuating unfolded state characterized by extended regions that persist up until the transition state is reached. It then switches to a series of conformations with a partial, native-like structure, as indicated by the emergence of a persistent darker (red) patch (i.e., a nucleus of NCs). After that, there is a rapid transition from the partially folded intermediate state to more compact native-like conformations, i.e., Q~0.80. This transition corresponds to the collapse of fluctuating internal loops in the protein. In the final stages of the folding pathway, external loops at the surface of the protein rearrange to attain the final folded structure.

Our results are entirely consistent with previously reported experimental studies which indicate that a few residues, close in sequence, form a nucleus randomly which then serve as a template for the growth of the rest of the structure.[39] This nucleus is formed by those residues that have more chances to be in contact in the transition state. These studies have also found that the replacement of residues within the core by ones that lower its stability does not prevent the folding, but merely slows down folding kinetics. Therefore, the nucleus is simply a region of the protein that has a higher probability to form a native structure early and that this region has some variability. This observation is consistent with the variability of the nuclei we observe and the fact that for a protein confined in a 6σ 3-D cage we observed a TS with a smaller $<Q>_{TSE}$ than that for the bulk and the 10σ 3-D cage systems. In this high confinement case, the folding pathway follows through a different transition intermediate, which offers a greater thermodynamic stability to the folded state and leads to a faster kinetics at $T_f = 0.27$. Nevertheless, these TS intermediates contain some key native contacts for the formation of the core, which are also observed in the bulk and 10σ 3-D cage system.

V. Discussion and conclusions

In this paper, we have demonstrated that FFS-type simulations with a lattice protein model provide a dramatic improvement in efficiency over brute-force simulations for estimation of the transition rate constant and the sampling of transition paths. Furthermore, this sampling algorithm allows the determination of the TPE, which can be used to obtain the TSE and the properties that characterize this ensemble. Details of the folding mechanism can thus be obtained from the TSE with little extra effort.

For the lattice protein system studied here, the simulation of folding kinetics was preceded by a complementary study of the thermodynamic behavior for the range of temperatures where the native state is stable ($T<T_f$). Based on the analysis of C_v vs. T plot and the ΔU and $T\Delta S$ contributions to ΔA_{UF}, we found (as in previous studies) that confinement limits the conformational space available to the unfolded state, reducing its entropy. However, this comes with an enthalpic penalty that is lower in magnitude than the favorable contribution of the entropy change, resulting in a net stabilization of the confined folded protein. We then studied the kinetics of the system by determining the average transition rate constant for the folding process. It has previously been reported that the confinement of a protein inside cylindrical and spherical cages can retard or enhance its folding kinetics depending on the temperature and the confinement dimensions.[7,8,12,37] For instance, Kirmizialtin et al. (2004) studied the translocation of a β-hairpin through a cylindrical tunnel that mimics the exit tunnel in a ribosome.[3] They argued that as a protein chain moves through a tunnel it goes through a series of compact folding-intermediates that could retard or accelerate its folding kinetics, depending on the tunnel's size. Contreras et al. also showed that acceleration in folding resulted from the pre-formation of a subset of native contacts for temperatures exceeding T_{opt} when a protein (the same one adopted in this study)

translocates through a tunnel having a cross section above a certain threshold.[12] Our results confirm that the kinetics can be accelerated for $T > T_{opt}$ where (for a given T) k_{UF} increases with the number of constrained dimensions. However, strong confinement is detrimental to folding kinetics, because the protein is trapped in local minimum energy conformations, where the global reconfiguration needed to attain its ground state is difficult.

For two-state proteins as the one studied here, there is just one single rate-determining transition state to be considered between the folded and unfolded states. Consequently, the mechanistic study of the folding pathways was focused on the statistical characterization of the transition state properties. The distribution of order parameter in the TSE (e.g., fractional nativeness) revealed that this model lattice protein folds to a unique native structure through a folding pathway that is characterized by cooperative transitions. These transient kinetic intermediates resemble equilibrium fluctuations from the native state, which are characterized by a strongly localized nucleus. This core of native contacts consists of the regions with higher probabilities to fold and acts as foci for folding. This is consistent with results obtained from simulations of a lattice model 48-mer protein where a Go-type interaction scheme was adopted.[40] In that work, the details of the folding event were investigated by performing hundreds of independent brute force simulations and extracting their common features; protein folding was envisioned as an intra-molecular chemical reaction through a sequence of transiently populated intermediates, separated by cooperative transitions. We also observed that this nucleus is simply the set of those residues that are more readily formed in the transition state and it can vary depending on the degree of confinement of the system.

Although lattice models do not capture all features that are critical for understanding protein folding, they do capture certain essential elements of the

thermodynamic and kinetics effects of confinement. We envision that FFS schemes will be a valuable tool to understand the folding mechanism of more realistic all-atom protein models simulated with molecular dynamics and will be a complement to experimental biophysical tools (i.e. NMR spectroscopy), which can be extremely challenging.[41]

Acknowledgments

One of the authors (E.E.B.) is grateful to the Sloan Foundation for a graduate Fellowship and to the Department of Education for a GAAN Fellowship. Additional support from the National Science Foundation Award BES-0093769 is acknowledged. The authors are grateful to Dr. C.R.A. Abreu for his help with the free energy simulation codes.

Appendix: Transition Path Sampling (TPS) algorithm

1. TPS's Theory

A detailed description of TPS can be found in Refs. 17-19. In the TPS method, the transition rate constant between two stable states A and B is characterized by a correlation function of state populations in time, $C(t)$. If the rare transitions between the stable states A and B are separated by a single dynamical bottleneck, $C(t)$ can be define as the conditional probability to find the system in final region B at time t provided it started in A at time t=0. Consequently, once $C(t)$ is determined for times greater than the characteristic time required to cross the dynamical bottleneck, $k_{A\to B}$ can be extracted by taking the derivate of the correlation function $C(t)$:

$$k_{A\to B} \approx \frac{dC(t)}{dt} = \frac{d\langle h_A(x_0)h_B(x_t)\rangle / dt}{\langle h_A(x_0)h_B(x_{t'})\rangle} \times C(t') \qquad \text{(A1)}$$

where $\langle ... \rangle$ denote equilibrium ensemble averages and x_t is the set of coordinates specifying the state of the system at time t. The characteristic functions h_A and h_B indicate if the system is in region A or B, respectively: $h_{A,B}(x) = 1$ if $x \in A, B$ and is zero otherwise. The conditional probability $C(t)$ can be obtained from path sampling:

$$C(t) = \int_{\lambda^B_{\min}}^{\lambda^B_{\max}} d\lambda P(\lambda, t) \qquad \text{(A2)}$$

where $P(\lambda,t)$ is the probability to find a time slice x_t at an order parameter λ. $\lambda^B_{\min}$ and $\lambda^B_{\max}$ are the lower and upper bounds of λ at region B. $P(\lambda,t)$ is usually determined by umbrella sampling, in which the phase space is divided into a sequence of N+1 overlapping regions B(i) or "windows" such that B(0) = basin B and the union $\bigcup_{i=0}^{N} B(i)$ of all regions comprises the whole phase space. The regions B(i) are defined through $x \in B(i)$ if $\lambda^{(i)}_{\min} \le \lambda(x) \le \lambda^{(i)}_{\max}$. The distribution of λ in each window, $P(\lambda,t;i)$, is then obtained by path sampling with the final region defined by B(i) and making a histogram of $\lambda(x_t)$. $P(\lambda,t)$ is obtained by matching the histograms $P(\lambda,t;i)$ in the overlapping regions and normalizing the resulting distribution. Finally, $C(t)$ is determined by integration of $P(\lambda,t)$ over values of λ corresponding to the final region B. So, $C(t)$ in the time interval $0 < t < \tau$ is determined by: (i) a single path sampling simulation to calculate $\langle h_A(x_0) h_B(x_t) \rangle$, and (ii) an umbrella simulation to get $C(t')$ for $t' << \tau$.

2. Rate constant results from TPS method

In Sec. IV B, we compared estimates of k_{UF} using different methods. Here, we describe how the TPS simulations were performed for the estimation of k_{UF}. A trajectory of length τ was denoted by an ordered sequence of states, $X(\tau) \equiv \{x_0, x_{1\Delta t}, x_{2\Delta t}, ..., x_\tau\}$, which are separated by a time interval increment, Δt. In our simulations, a

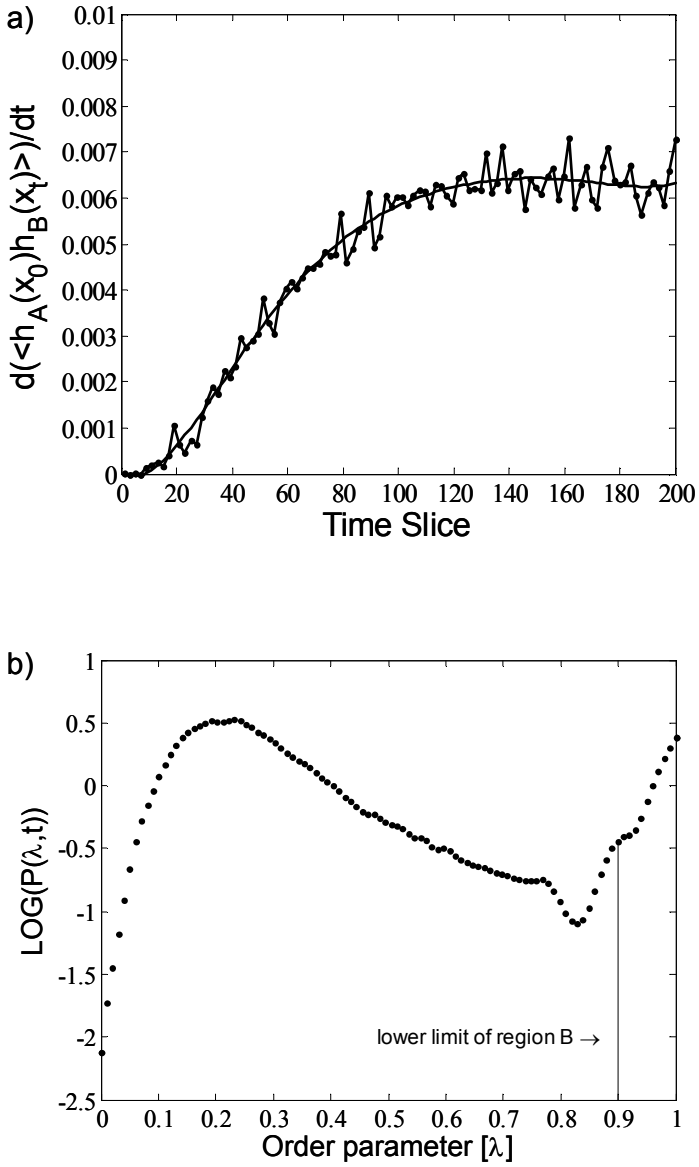

Figure 4.12 . (a) Time derivative $d\langle h_A(x_0)h_B(x_t)\rangle/dt$ along transition paths of length L=200 for a protein folding in the bulk at T_{opt}=0.23. (b) Logarithm of the probability to find the endpoint of the path of length L=100 at an order parameterλ. *P(λ,t')* is normalized such that $\int_0^\infty P(\lambda,t')d\lambda = 1$. The solid line bounded the final region B given by $0.9 \le \lambda(x) \le 1.0$.

representative trajectory connecting the unfolded and folded states consisted of 200 states, or time slices (ι), separated by a $\Delta t=3x10^3$ MC steps. The unfolded (A) and folded (B) regions were defined in terms of Q, i.e. $\lambda(x) = Q$. The parameter $\lambda(x)$ was fixed, such that the system has values $\lambda(x) \leq \lambda_A(x) = 0.1$ in region A and $\lambda(x) \geq \lambda_B(x) = 0.9$ in region B. The TPE was sampled by applying shooting moves. In a shooting move, a phase point $x_\Gamma^{(0)}$ is selected at random from the consecutive sequence of states along an old pathway, $X^{(0)}(\tau)$. Then, a new trajectory segment from Γ to τ is generated by applying the dynamical propagation rules to $x_\Gamma^{(0)}$. Thus, the new trajectory $X^{(n)}(\tau)$ consists of 0 to Γ states from the old trajectory and Γ to τ states from the new segment. If the new trajectory is reactive, i.e., if its initial state lies in region A and ends in B, the path is accepted and used as the current one. Otherwise the old trajectory is counted again in the calculation of path averages. The path sampling simulations were carried out as a series of blocks, each consisting of 1000 attempted shooting moves. The path averages were taken over all blocks.

The average $\langle h_A(x_0)h_B(x_t)\rangle$ in the interval $0 < t < \tau$ was determined from a single TPS run. In this simulation, pathways connecting the region A and B without requiring ending in B are sampled and collected as the TPE. Figure 4.12 (a) shows the time derivative $d\langle h_A(x_0)h_B(x_t)\rangle/dt$ as a function of time slices (i.e. $t = \iota \times \Delta t$). Clearly, $d\langle h_A(x_0)h_B(x_t)\rangle/dt$ reaches a plateau of 6.1×10^{-3} after $\iota \approx 140$. The correlation function $C(t')$ was calculated for an intermediate time slice $\iota'=100$ (i.e., $t = \iota' \times \Delta t$) under the same path sampling conditions. During the umbrella sampling, pathways connecting the regions A and B are required to end in B(i) to collect statistics for the histogram of the $\lambda(x_t)$. Figure 4.12 (b) shows the logarithm of the probability $P(\lambda,t')$ to find the endpoint at a certain λ. $P(\lambda,t')$ was determined from path sampling simulations for six overlapping windows: $0 \leq \lambda \leq 0.1$, $0.1 \leq \lambda \leq 0.2$, $0.2 \leq \lambda \leq 0.4$, $0.4 \leq \lambda \leq 0.6$, $0.6 \leq \lambda \leq 0.8$, and $0.8 \leq \lambda \leq 1.0$. For each window we performed 10

blocks of path sampling simulations, each consisting of 1000 attempted shooting moves in which the endpoint of the path was confined to the respective λ window. The histograms of λ for the different windows were stitched together by finding suitable multiplying constants. $P(\lambda,t')$ was normalized such that $\int_0^\infty P(\lambda,t')d\lambda = 1$. Integration of the normalized $P(\lambda,t')$ over the final region B [$\lambda \in (0.9,1.0)$] gives $C(t')$=0.0991. The value of $\langle h_A(x_0)h_B(x_{t'})\rangle$ was estimated to be 0.2935. Using Eq. (A1), we found k_{UF}= $2.06\text{x}10^{-2}$ (time slice)$^{-1}$ which once divided by the time interval between slices ($\Delta t=3\text{x}10^3$ MC steps) we got a value of $6.866\text{x}10^{-7}$ for the protein folding in the bulk at T_{opt}=0.23.

REFERENCES

[1] A. Sethuraman, G. Vedantham, T. Imoto, T. Przybycien, and G. Belfort, Proteins: Struct., Funct., Bioinf. **56**, 669 (2004).

[2] K. Tsumoto, D. Ejima, I. Kumagai, and T. Arakawa, Protein Expression Purif. **28**, 1 (2003).

[3] S. Kirmizialtin, V. Ganesan, and D. E. Makarov, J. Chem. Phys. **121**, 10268 (2004).

[4] B. Schimmele, N. Grafe, and A. Pluckthun, Protein Eng., Des. Sel. **18**, 285 (2005).

[5] H. P. Sorensen, J. E. Kristensen, H. U. Sperling-Petersen, and K. K. Mortensen, Biochem. Biophys. Res. Commun. **319**, 715 (2004).

[6] C. D. Sfatos, A. M. Gutin, V. I. Abkevich, and E. I. Shakhnovich, Biochemistry **35**, 334 (1996).

[7] A. Baumketner, A. Jewett, and J. E. Shea, J. Mol. Biol. **332**, 701 (2003).

[8] M. R. Betancourt and D. Thirumalai, J. Mol. Biol. **287**, 627 (1999).

[9] T. H. Cho, S. J. Ahn, and E. K. Lee, Bioseparation **10**, 189 (2001).

[10] A. I. Jewett, A. Baumketner, and J. E. Shea, Proc. Natl. Acad. Sci. U. S. A. **101**, 13192 (2004).

[11] G. Ziv, G. Haran, and D. Thirumalai, Proc. Natl. Acad. Sci. U. S. A. **102**, 18956 (2005).

[12] L. M. Contreras, F. Martinez-Veracoechea, P. Pohkarel, A. D. Stroock, F. Escobedo, and M. P. DeLisa, Biotechnol Bioeng **94**, 105 (2006).

[13] R. J. Allen, D. Frenkel, and P. R. ten Wolde, J. Chem. Phys. **124**, 194111 (2006).

[14] R. J. Allen, D. Frenkel, and P. R. ten Wolde, J. Chem. Phys. **124**, 024102 (2006).

[15] D. Chandler, *Introduction to Morden statistical Mechanics*. (Oxford University Press, New York, 1987).

[16] D. Frenkel and B. Smit, *Understanding Molecular Simulation: From Algorithms to Applications*, 2nd ed. (Academic, Boston, 2002).

[17] G. E. Crooks and D. Chandler, Phys. Rev. E **6402**, 026109 (2001).

[18] P. G. Bolhuis, Proc. Natl. Acad. Sci. U. S. A. **100**, 12129 (2003).

[19] P. G. Bolhuis, D. Chandler, C. Dellago, and P. L. Geissler, Annu. Rev. Phys. Chem. **53**, 291 (2002).

[20] C. Dellago, P. G. Bolhuis, F. S. Csajka, and D. Chandler, J. Chem. Phys. **108**, 1964 (1998).

[21] C. Dellago, P. G. Bolhuis, and P. L. Geissler, Adv. Chem. Phys. **123**, 1 (2002).

[22] A. K. Faradjian and R. Elber, J. Chem. Phys. **120**, 10880 (2004).

[23] T. S. van Erp and P. G. Bolhuis, J. Comput. Phys. **205**, 157 (2005).

[24] T. S. van Erp, D. Moroni, and P. G. Bolhuis, J. Chem. Phys. **118**, 7762 (2003).

[25] D. Moroni, T. S. van Erp, and P. G. Bolhuis, Physica A (Amsterdam) **340**, 395 (2004).

[26] R. J. Allen, P. B. Warren, and P. R. ten Wolde, Phys. Rev. Lett. **94**, 018104 (2005).

[27] Z. Bagci, R. L. Jernigan, and I. Bahar, Polymer **43**, 451 (2002).

[28] S. Miyazaka and R. Jernigan, Macromolecules **18**, 543 (1985).

[29] K. Leonhard, J. M. Prausnitz, and C. J. Radke, Phys. Chem. Chem. Phys. **5**, 5291 (2003).

[30] V. I. Abkevich, A. M. Gutin, and E. I. Shakhnovich, Fold Des. **13**, 221 (1996).

[31] H. Kaya and H. S. Chan, Phys. Rev. Lett. **85**, 4823 (2000).

[32] M. Friedel, D. J. Sheeler, and J. E. Shea, J. Chem. Phys. **118**, 8106 (2003).

[33] D. Thirumalai, D. K. Klimov, and G. H. Lorimer, Proc. Natl. Acad. Sci. U. S. A. **100**, 11195 (2003).

[34] D. J. Earl and M. W. Deem, Phys. Chem. Chem. Phys. **7**, 3910 (2005).

[35] A. M. Ferrenberg and R. H. Swendsen, Phys. Rev. Lett. **63**, 1195 (1989).

[36] S. Trebst, M. Troyer, and U. H. E. Hansmann, J. Chem. Phys. **124**, 174903 (2006).

[37] F. Takagi, N. Koga, and S. Takada, Proc. Natl. Acad. Sci. U. S. A. **100**, 11367 (2003).

[38] A. R. Fersht, Proc. Natl. Acad. Sci. U. S. A. **92**, 10869 (1995).

[39] A. R. Fersht, Curr. Opin. Struct. Biol. **7**, 3 (1997).

[40] V. S. Pande and D. S. Rokhsar, Proc. Natl. Acad. Sci. U. S. A. **96**, 1273 (1999).

[41] H. X. Zhou and K. A. Dill, Biochemistry **40**, 11289 (2001).

5. CHAPTER: Kinetics and reaction coordinates of the reassembly of protein fragments via forward flux sampling*

Abstract

We study the mechanism of the reassembly and folding process of two fragments of a split lattice protein by using forward flux sampling (FFS). Our results confirm previous thermodynamics and kinetics analyses that suggested that the disruption of the critical core (of the unsplit protein that folds by a nucleation mechanism) plays a key role in the reassembly mechanism of the split system. For several split systems (from a parent 48-mer model), we estimate the reaction coordinates in term of collective variables by employing the FFS-least square estimation method and found that the reassembly transition is best described by a combination of the total number of native contacts in both chains, the number of interchain native contacts, and the conformational energy. We also analyzed the transition path ensemble obtained from FFS simulations using the estimated reaction coordinate as order parameter to identify the microscopic features that differentiate the reassembly of the different split systems studied. We found that in the faster folding systems a balanced distribution of the original-core amino acids between protein fragments propitiates inter-chain interactions at early stages of the folding process. In the slower folding system, the concentration of the folding nucleus in one fragment causes its early prefolding while the second fragment tends to remain as a detached random coil. For such a slower folding system, a single point mutation that

* Borrero, E. E.; Contreras-Martinez, L. M.; DeLisa, M. P.; and Escobedo, F. A. Biophysical Journal. (in preparation).

strengthens one of the native interchain contacts (prevalent in the transition state ensemble) is shown to enhance interchain cooperativeness and the rate of reassembly.

I. Introduction

Several transition path sampling (TPS) methods have been developed to study the kinetics of rare events at a molecular level. These approaches (1-11) allow the computation of rate constants of rare events by enhancing the sampling of the phase space in the transition region. Moreover, these methods can generate the collection of pathways [i.e., the transition path ensemble (TPE)] and give mechanistic details of the system's dynamics between states, which would be impractical to obtain by conventional brute-force simulation techniques. The path sampling technique of choice for this study is the Forward Flux Sampling method because of its simplicity and efficiency. For a comprehensive summary of applications of different path sampling methods to biological systems, the readers are referred to Refs. (12, 13).

Recently, two of us used FFS to evaluate the kinetics of the transition pathways for the folding mechanism of a single chain (unsplit) protein (14), based on coarse-grained lattice models. That work showed that the initial formation of a critical core of amino acids is the most important step during folding, a result that is relevant for proteins following a nucleation folding mechanism. The critical core is comprised by those residues that have a higher chance to form a native contact in the transition state. (15, 16) In a follow-up paper, an approach that combines FFS with a least square estimation (LSE), denoted as FFS-LSE, was used for estimating a good reaction coordinate for the folding transition of this lattice protein, thus identifying the main variables that described the intrinsic transition dynamics. (17)

In a previous study, we used brute-force Monte Carlo simulations to investigate the thermodynamics and kinetics of the reassembly process of the two

fragments of a split lattice protein; in particular, to elucidate how the disruption of the folding nucleus affects the reassembly and folding of the fragments depending on the location of the split site. (18) With the help of a thermodynamic analysis, it was argued that the balanced distribution of the amino acids that form the folding core between protein fragments was key to their reassembly. In this work we employ the FFS method to simulate the TPE and to try to attain a more detailed and quantitative picture of how the reassembly process differs for different split site locations and how it relates to the folding of the unsplit protein. Our strategy is to take full advantage of an adaptive optimization algorithm for FFS-type approaches (19) to efficiently implement FFS-LSE simulations (17) to obtain a good estimate for the reaction coordinate. Note that the reaction coordinate essentially corresponds to the so-called committor probability (p_B) surface, which gives the probability of any particular system configuration to reach the final folded state. After the reaction coordinate is parameterized in terms of collective variables (i.e., macroscopic, physically meaningful properties) that describe the system's dynamic, the mechanistic details of the transition can be extracted by screening the microscopic properties of the ensemble of configuration belonging to different regions of the committor probability surface (e.g., for p_B =0.5 which corresponds to the transition state).

The article is organized as follows. After reviewing the FFS-type methods and the simulation details for the split systems (including characterization of the fragments), we report the results for the reaction coordinate models and discuss the mechanism for the reassembly of all the split systems studied. We end by summarizing our main results and providing an outlook for future studies.

II. Methods

A. Forward Flux Sampling (FFS)

We employ the Branched Growth (BG) sampling scheme to generate the transition path ensemble (TPE). (1-3) The algorithm also provides a simple expression for the calculation of the average rate of transitions $k_{A\to B}$ from two well-defined states A and B using an "effective positive flux" expression:(1-3, 9)

$$k_{A\to B} = \overline{\Phi}_{A,0} P(\lambda_{n=B} \mid \lambda_0) \ . \qquad (1)$$

The $\overline{\Phi}_{A,0} = N_0 / \Gamma$ term is the total average flux of trajectories from A to λ_0, which is calculated by performing a simulation of time Γ in the initial basin A and counting the number N_0 of effective positive crossing events of the first interface at λ_0. $P(\lambda_{n=B} \mid \lambda_0)$ is the probability that a trajectory reaching λ_0 from A will reach to B without returning to A. To enhance the sampling and obtain estimates for $P(\lambda_{n=B} \mid \lambda_0)$ the phase space is partitioned by employing a series of nonintersecting interfaces (n+1) such that the system is considered in region A for $\lambda(x) \le \lambda_A(x)$ and in region B for $\lambda(x) \ge \lambda_B(x)$. The interfaces are defined by an order parameter $\lambda(x)$ (where x is the phase space coordinates) whose values increase monotonically as the interfaces come close to region B. Starting from a random selected configuration at λ_0, a branched "tree" is generated by firing k_i trial runs for each configuration reaching successive interfaces λ_i ($0 \le i \le n$-1). If one estimates the conditional probability to jump from one interface to the next as

$$P(\lambda_{i+1} \mid \lambda_i) = \frac{N_s^i}{N_s^{(i-1)} k_i} \qquad (2)$$

where N_S^j is the number of branches fired from interface j that reached the next interface, then $P(\lambda_{n=B} \mid \lambda_0)$ can be found from the product of such conditional probabilities:

$$P(\lambda_{n=B} \mid \lambda_0) = \prod_{i=0}^{n-1} P(\lambda_{i+1} \mid \lambda_i) = \frac{N_S^{(n-1)}}{\prod_{i=0}^{n-1} k_i} \qquad (3)$$

which reduces to the calculation of the ratio of the total number $N_S^{(n-1)}$ of branches that eventually reach λ_n, to the total possible number of branches ($\prod_{i=0}^{n-1} k_i$). For a complete description of the theoretical basis and implementation details of the algorithm, see Ref. (2, 3).

B. FFS-LSE algorithm

The effectiveness and usefulness of the FFS algorithm depends on the λ order parameter chosen to describe the transition and the partitioning of this λ space between the stable states. (17) The FFS-LSE protocol extracts "on-the-fly" B-committor probability (p_B history) data from a BG simulation that used an initial guess for the order parameter to obtain an improved order parameter or reaction coordinate. The p_B value for every interfacial point stored in the TPE trajectories is obtained by using a recursively equation:

$$p_{B\,j}^{\ i} = \frac{\sum_{m=1}^{N_j^i} p_{B\,m}^{\ i+1}}{k_i}, \qquad i=n\text{-}1,n\text{-}2,\ldots,1 \quad (4)$$

where $p_{B\,m}^{\ i+1}$ is the committor probability to reach B for each point m at λ_{i+1} that connect with the state j at λ_i. Because p_B is the ideal reaction coordinate, the optimized order-parameter model will be one that is able to "fit" well the p_B history data. (17) To

estimate such a reaction coordinate model, the p_B history data collected over the whole phase space region connecting states A and B is fitted to a mathematical relation that depends on any number of candidate collective variables (suspected to be meaningful order parameters for the system's dynamics). Standard least-square estimation (LSE) and an analysis of variance (ANOVA) are performed to find the statistically significant terms in the model. The readers are referred to Ref. (17) for a detailed description of the FFS-LSE method.

C. Adaptive λ staging optimization algorithm

The efficiency of the FFS simulation also depends on the λ staging. The adaptive optimization algorithm from Ref. [19] seeks to reposition the interfaces $\{\lambda\}$ to allocate the computational effort in a FFS simulation in such a way as to concentrate the sampling in the bottleneck regions. This optimization procedure states that for optimal sampling, the $P(\lambda_{i+1} \mid \lambda_i)$ values [which are determined by the $\{\lambda\}$ values] must be set to attain a net constant flux of partial trajectories between interfaces. To do this, initial estimates for the $P(\lambda_{i+1} \mid \lambda_i)$ probabilities [from Eq. (2)] for an arbitrary initial choice of $\{\lambda\}$ are fed back to construct a special "interpolating" function f that provides a one-to-one correspondence between an f value and a λ value [via Eq. (40) in Ref. (19)]. The new $\{\lambda'\}$ staging is thus obtained from the f values that correspond to desired $P(\lambda_{i+1} \mid \lambda_i)$ values. Note that in this work, we use a combined optimization strategy wherein after the $\boldsymbol{\lambda}$ phase staging is optimized to position interfaces at the TSE and "bottleneck" regions, the $\{k_i\}$ set is then optimized [via Eq. (31) in Ref. (19)] to improve the sampling of those important regions for a fixed computational cost.

Given that the FFS-LSE method requires a comprehensive p_B history, the optimized $\{\lambda_i'\}$ set was obtained by distributing the $P(\lambda_{i+1} \mid \lambda_i)$ values trying to target prescribed values of $\langle p_B \rangle_\lambda$ according to,

$$P(\lambda_{i+1} \mid \lambda_i) = \langle p_B \rangle_{\lambda_i} / \langle p_B \rangle_{\lambda_{i+1}} \qquad (5)$$

where $\langle p_B \rangle_{\lambda_i}$ is the "desired" average p_B value for all points at λ_i. Note that the $\langle p_B \rangle_\lambda$ isosurfaces prescribed in Eq. (5) are only approximate but will approach the true p_B isosufaces if the order parameter used is a good reaction coordinate.

III. Simulation details

A. Unsplit system

The 48-mer protein model adopted here exhibits a fast and stable proteinlike folding into a unique native structure via a two-state (unfolded-folded) process whose transition pathways are known to follow a nucleation-driven folding mechanism. (14) The formation of a critical core of amino acids mediates the folding of the single-chain (unsplit) protein. This critical core is formed at an early stage of the process by those residues that have a higher chance of being in contact in the transition state. (15, 16) Borrero and Escobedo (14) reported that this nucleus is composed of several (mostly hydrophobic) amino acid residues, that have over 80% probability of forming native contacts (i.e., residues: 13, 16-17, 19-24, 26-31 and 34-47); forming a core at the center of the folded structure. Figure 5.1 shows the contact map density for all the native contacts belonging to the ensemble of configurations at isocommittor p_B = 0.2, 0.5 and 0.8 surfaces. The configurations belonging to different regions of the

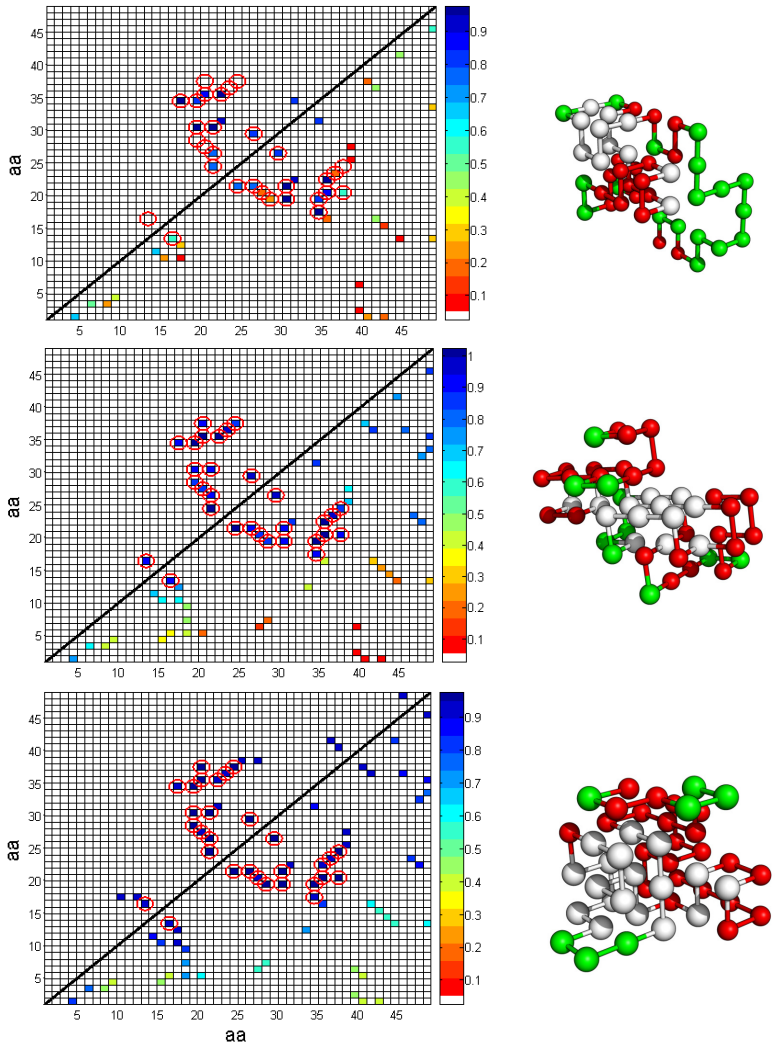

Figure 5.1 Contact density map for the 48-mer system for ensembles of configurations belonging to isocommittor surfaces: p_B=0.2 (top), p_B=0.5 (center), and p_B=0.8 (bottom). The x- and y-axis represent the amino acid (aa) position in the 48-mer sequence. The ensembles were collected by estimating p_B values for all the interfacial points in the TPE from the reaction coordinate model in Eq. (6). The lower triangle (below the diagonal line) shows the probability of a native contact to belong to the ensemble; the color code is given by the vertical bar. The upper triangle shows those native contacts with at least 80% probability to belong to the corresponding p_B ensemble. Encircled symbols represent native contacts that form the critical folding nucleus. Snapshots depicting typical configurations observed for each ensemble are also shown where red indicates intrachain native contacts and white indicates native contacts that form the critical folding nuclei.

isocommittor surface were collected by calculating the p_B value for each TPE interfacial configuration via the following equation:

$$p_B(NC, E) = -0.404 + 0.017(NC) - 0.029(E). \qquad (6)$$

This reaction coordinate model projects the committor probability (p_B) surface on the phase space, i.e., it estimates the probability of any TPE interfacial point to commit to the folded state from the *NC* and *E* values of that point.(17) Figure 5.1(b) shows that the 15 critical core (CC) native contacts with higher probability to belong to the transition state (p_B = 0.5) are formed by those residues forming the folding nucleus. Figure 5.1(a) reveals that these CC native contacts start to form the nucleus at early stages of the folding process (p_B = 0.2). Figure 5.1(c) shows that at late stages of the folding process (p_B = 0.8), the protein acquires its native structure by forming contacts around the folding nucleus.

Recently, we hypothesized and demonstrated that the way in which this nucleus is fragmented plays a critical role in the reassembly of split proteins. (18) In the present work, we use the same split protein systems to study in detail the kinetics and mechanism of the reassembly process.

B. Split System Preparation

The split lattice model proteins were generated by dissecting the 48- mer at one of three possible positions: between residues 16 and 17 (N-split), 24 and 25 (M-split), and 32 and 33 (C-split). In all these cases the folded state is identical to the one reached by the single 48-mer chain and characterized by NNC =58 native contacts. Table 5.1 gives the main characteristics for all the systems (i.e., the split and the unsplit systems), including the number of *inter*-chain CC contacts (InterC) that involve interacting residues from both chains and *intra*-chain CC contacts (IntraC) that

Table 5.1 Characteristics of the 48-1 mer and split systems. NNA and NNB are the number of intrachain contacts formed in chain A and B, respectively. INC corresponds to the number of native contacts formed between the two fragments. The distribution of the 15 native contacts forming the folding nuclei in split-proteins: *inter*-chain contacts (InterC) that involve interacting residues from both chains and *intra*-chain contacts (IntraC) that involve interacting residues within the same chain.

	Number of residues		**NC**	**Energy** $[k_BT]$	**NNA**	**NNB**	**INC**	**InterC**	**IntraC**	
	chain A	**chain B**							**chain A**	**chain B**
48- mer	48	---	57	-20.24	57	---	---	---	15	---
C-split	32	16	58	-20.62	25	7	26	7	8	---
M-split	24	24	58	-20.65	16	12	30	13	2	---
N-split	16	32	58	-20.43	9	28	21	---	1	14

involve interacting residues within the same chain.(18) These InterC and IntraC are formed by the "original" 15 critical core native contacts (identified in Figure 5.1) upon protein fragmentation.(14) Note that in the C- and Mid-split cases, the folding core residues are well distributed between fragments and give a significant number of InterC. In contrast, for the N-split system most of the folding core residues are concentrated in chain B and are not involved in InterC. Moreover, the N-split and the C-split systems are symmetrical, with each system having one 16-mer fragment and one 32-mer fragment; this allows a comparison in the absence of chain length disparities. Further details on the model unsplit and split systems, including their structure and thermodynamics are given in Ref. (18).

C. Conformational Sampling

Conformational local sampling was performed through a set of MC moves based on the Verdier-Stockmayer algorithm. (20) Relative to these local moves, whole-fragment diffusional translation of a randomly selected chain was also attempted after each MC step with a priori probability ($\leq 10^{-4}$). (18) For simulating the folding kinetics, the temperature was fixed at T=0.25, a value close to the folding transition temperature of the unsplit system. The system was confined inside a relatively large 3D cubic box of side length 12 σ (where σ is the lattice size = size of a protein residue) corresponding to a protein volume fraction of ~3%. (18)

In comparing different split protein systems, the analysis in our previous work (18) indicated that differences in kinetic behavior were not determined by diffusion limitations of the fragments trying to find each other. The spatial constriction adopted was intended to imitate a moderately crowded environment relative to open space, ensuring a timely association of the different fragments.

D. Candidate Collective Variables

In the simulations, the following macroscopic properties were calculated for all the state points collected at the λ interfaces in the TPE trajectories: total number of native contacts (NC), number native contacts in chain A (NNA), number native contacts in chain B (NNB), number of contacts between fragments (IC), number of native contacts between fragments (INC), conformational energy (E), and the number of critical core contacts (CC) (the latter as identified for the unsplit system). These collective variables were used for the reaction coordinate analysis via the FFS-LSE method.

IV. Results

A first preliminary BG simulation was performed using the number of native contacts as initial guess of the order parameter (i.e., λ=*NC*) with the purpose of optimizing the position (λ values) and sampling (k values) of 12 interface ensembles. Details of this calculation and its results are given in the supplementary material. These optimized $\{\lambda_i'\}$ and $\{k_i'\}$ sets were then used to obtain the p_B history data via BG simulations with the FFS-LSE method. Ten blocks were used where each block consisted of 5 BG runs started from a randomly selected configuration at λ_0 from which a branched path is generated and then used to estimate committor probabilities p_B. These p_B data were then used to screen a set of candidate collective properties (see Sec. III D) for an optimized order parameter model λ, as described in Sec. II B. Thereafter, a better estimate for the order parameter is obtained and is used to partition the phase space for additional FFS-LSE simulations. The adaptive optimization algorithm is also applied to find a better $\boldsymbol{\lambda}$ staging of the new order parameter between iterations. This combination of FFS-LSE and staging optimization provides the advantage of allowing a more efficient and uniform distribution of the p_B data over the

Table 5.2 FFS-LSE parameters for the reaction coordinate model [$p_B \approx \beta + \beta_1 NC + \beta_2 INC + \beta_3 E + \beta_4 NC^2 + \beta_5 INC^2 + \beta_6 E^2 + \beta_7 NC \times INC + \beta_8 NC \times E + \beta_9 INC \times E$]. The significance of any individual regression coefficient for the model description is indicated by the partial $F_0 = MS_{SSR} / MS_E$ value; i.e., the ratio of the regression sum square due to β_j and the mean square for the residuals. For any nonzero regression term, the *P*-value for the F_0 statistics is less than α (here chosen to be to 0.05 for a 95% confidence interval). The β_j significance increases with the F_0 value.

System	Model	Model coefficient (β) [F_0]									
		(β_0)	*NC*	*INC*	*E*	NC^2	INC^2	E^2	*NC*×*INC*	*NC*×*E*	*INC*×*E*
N-split	1	-0.75	--	0.083 [14913]	--	--	--	--	--	--	--
	2	-1.78	0.069 [27]	-0.038 [108]	-0.095 [19]	-0.003 [147]	-0.003 [179]	-0.019 [119]	-0.002 [32]	-0.005 [40]	-0.01 [90]
Mid-split	1	-0.43	--	0.048 [1707]	--	--	--	--	--	--	--
	2	-0.86	0.019 [16]	0.018 [53]	-0.043 [158]	-0.001 [30]	-0.005 [333]	-0.002 [166]	0.005 [538]	--	--
C-split	1	-0.82	--	0.079 [2020]	--	--	--	--	--	--	--
	2	-1.87	0.060 [13]	0.016 [53]	-0.084 [12]	-0.002 [82]	-0.009 [777]	-0.010 [22]	0.005 [140]	-0.006 [28]	-0.003 [17]

entire phase space. The combined scheme is repeated until the FFS-LSE is fully converged, i.e., when similar reaction coordinate models are obtained in consecutive iterations.

Table 5.2 shows the coefficients for the reaction coordinate models obtained from the iterations of the combined scheme (i.e., FFS-LSE and adaptive optimization algorithm). For the first iteration, the LSE and ANOVA for this reaction coordinate model (i.e., model 1 in Table 5.2) indicates that the number of native contact between fragments (INC) is the only and most significant collective variable which describes the system transition to the folded state. Note that the number of native contacts (NC) turned out to be a very poor choice of order parameter to describe the reassembly transition for the split systems, while a linear model in INC should work better. A second iteration was then performed using this new estimate of order parameter and n=9 interfaces initially positioned at λ_i ($0 \le i < n$): $\boldsymbol{\lambda}(x,y)$ = {0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8}. The stables states A and B were defined as before. This initial λ staging was subsequently optimized by distributing the $P(\lambda_{i+1} \mid \lambda_i)$ values trying to target prescribed values of $\langle p_B \rangle_\lambda$ according to Eq. (5) with $\langle p_B \rangle_{\lambda_i}$ = {0.4, 0.5, 0.6, 0.8} for 5 $\le \lambda_i \le n$-1, and $P(\lambda_{i+1} \mid \lambda_i) = P_\lambda$ for $0 \le \lambda_i \le 4$, where $P_\lambda = \left[P(\lambda_{n=B} \mid \lambda_0) / \prod_{i=1}^{n-1} P(\lambda_{i+1} \mid \lambda_i)\right]^{1/5}$ = constant. In this way, the λ interfaces are distributed over the entire phase space region: the uphill region close to basin A (λ_0 - λ_4), the transition state (TS) region (λ_5 -λ_7), and the region close to basin B (λ_8 and λ_9).

Again the model surfaces were initially fitted to the set of candidate collective variables (see Sec. III D) and ANOVA predicted significant linear, quadratic and interaction terms for the variables INC, NC and E. A second LSE fitting was then performed only considering these three collective variables to obtain the reaction coordinate model 2 in Table 5.2. The initial and optimized {λ_i'} sets for model 2 of all split systems are given in Table 5.3 (note also that we fixed k_i=10). The p_B surfaces

Table 5.3 Optimized $\{\lambda'\}$ set for the model 2 reaction coordinate in Table 5.3. The $\langle p_B \rangle$ values were approximated by $\langle p_B \rangle^{\lambda_{n-1}} = P(\lambda_{n=B} \mid \lambda_{n-1})$ for i=n-1 and $\langle p_B \rangle^{\lambda_{n-i}} = P(\lambda_{n-i+1} \mid \lambda_{n-i}) \langle p_B \rangle^{\lambda_{n-i+1}}$ for $0 \le i \le n$-2.

i	Initial $\{\lambda\}$ set	C-split			Mid-split			N-split		
	λ_i	λ_i'	$P(\lambda_{i+1}\|\lambda_i)$	$\langle p_B \rangle$	λ_i'	$P(\lambda_{i+1}\|\lambda_i)$	$\langle p_B \rangle$	λ_i'	$P(\lambda_{i+1}\|\lambda_i)$	$\langle p_B \rangle$
0	0.0	0.00	0.83	0.12	0.00	0.54	0.05	0.00	0.74	0.09
1	0.1	0.06	0.71	0.14	0.07	0.75	0.08	0.06	0.77	0.11
2	0.2	0.11	0.83	0.20	0.13	0.67	0.11	0.12	0.75	0.15
3	0.3	0.18	0.81	0.24	0.20	0.60	0.17	0.18	0.74	0.20
4	0.4	0.26	0.73	0.30	0.27	0.64	0.28	0.23	0.78	0.27
5	0.5	0.38	0.82	0.41	0.35	0.84	0.43	0.38	0.76	0.35
6	0.6	0.48	0.78	0.50	0.49	0.72	0.51	0.46	0.75	0.46
7	0.7	0.61	0.76	0.65	0.59	0.79	0.71	0.59	0.73	0.61
8	0.8	0.70	0.85	0.85	0.70	0.90	0.90	0.81	0.83	0.83
9 (=n)	58	58	----	----	58	----	----	58	----	----

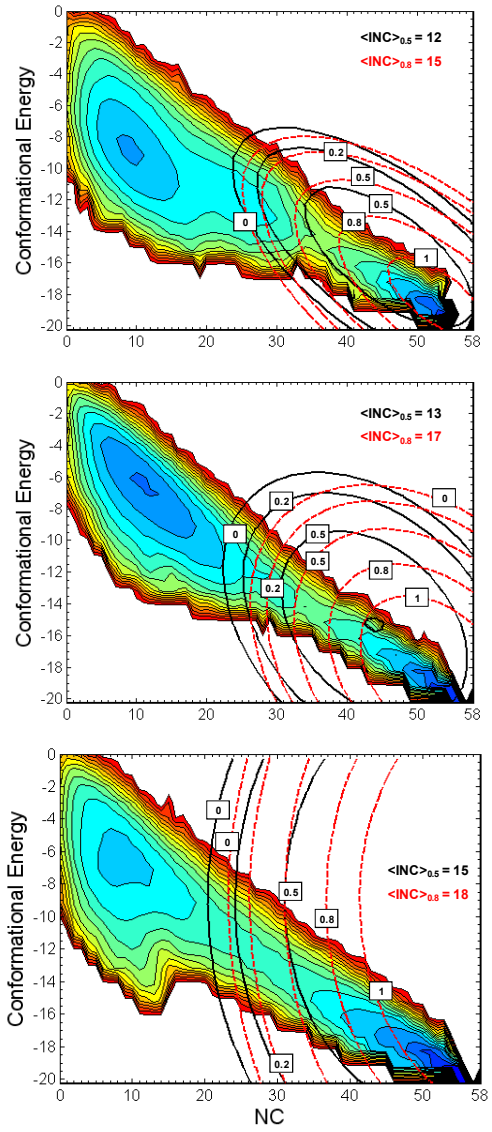

Figure 5.2 Estimated reaction coordinate iso-lines for all the split systems lattice protein models folding at T=0.25: N-split (top), C-split (center), and Mid-split (bottom). The lines are the predicted p_B committors from FFS-LSE (model 2 in Table 5.2) for fixed INC values corresponding to the ensemble average at p_B=0.5 (black lines), and p_B=0.8 (red dotted lines). The committor values appear as labels over the lines. Contour plot of the free energy landscape [energy (E) vs. nativeness (NC)] is shown only for visual reference. The color scheme changes from highest (red) to lowest (blue) elevations.

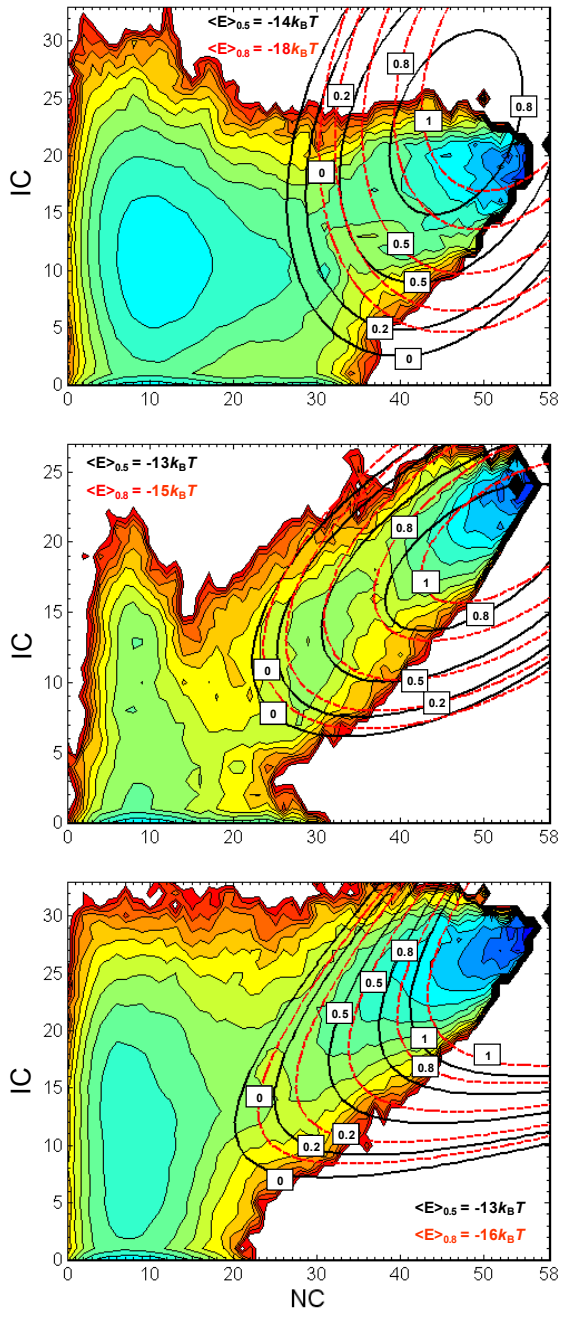

Figure 5.3 Estimated reaction coordinate iso-lines for all the split systems lattice protein models folding at T=0.25: N-split (top), C-split (center), and Mid-split (bottom). The lines are the predicted p_B committors from FFS-LSE (model 2 in Table II) for fixed conformational energy values corresponding to the ensemble average at p_B=0.5 lines (black lines), and p_B=0.8 (red dotted lines). The committor values appear as labels over the lines. Contour plot of the free energy landscape [interchain contacts (*IC*) vs. nativeness (*NC*)] is shown only for visual reference. The color scheme changes from highest (red) to lowest (blue) elevations.

predicted by the reaction coordinate of these models 2 are illustrated in Figures 5.2 and 5.3. Figure 5.2 projects the reaction coordinate model onto the conformational energy (E) and NC variables for fixed values of <INC> corresponding to the averages on the ensemble of configurations belonging to the surfaces p_B=0.5 and p_B=0.8. Figure 5.3 projects the reaction coordinate model onto the space of NC and the number of contacts between fragments (IC) for fixed values of <E> corresponding to averages on the ensemble of configurations belonging to the surfaces p_B=0.5 and p_B=0.8. Supplementary Figures 5.1-5.3 are intended to show graphically the qualitatively differences in how p_B is correlated by the significant variables (<INC>, < NC> and <E>) and the non-significant variables (i.e., <NNA>, <NNB>, <CC>, and <IC>) in the model.

Figure 5.4 compares the transition rate constants for the three split systems and the 48-mer folding at T=0.25. The folding rates for the N-, Mid- and C-split systems are ~ 12, 19, and 31% the rate of the 48-mer, respectively. More revealing, the N-split system folds at a rate that is 37% that of the C-split system (despite having fragments of equal lengths) and 61% that of the Mid-split system. In our previous study, these differences in folding kinetics were rationalized by comparing the differences in free energy barriers observed between the 48-mer and the different split systems. We found a shift in the transition state dividing surface especially for the N-split and Mid-split systems relative to the 48-mer transition, suggesting that the reassembly of these systems takes place via a different, slower folding mechanism.

Taking advantage of the availability of optimized order parameters, we can re-examine the thermodynamics of the split systems to search for clues that may explain the differences observed in their kinetic behavior. In particular, we define free energy (A), energy (E), and entropy (S) changes between the unfolded and folded states (ΔA,

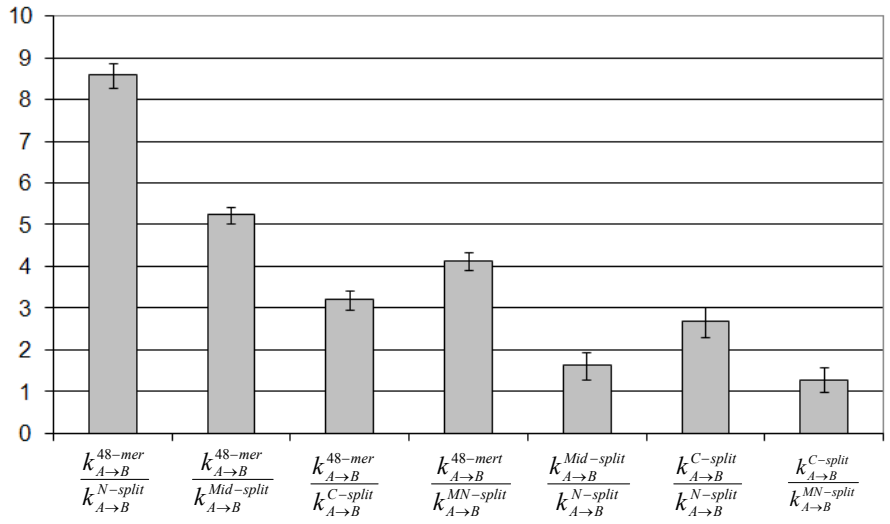

Figure 5.4 Ratio of the folding transition rate constants for all systems.

ΔE , and ΔS) and between the TS and the unfolded states (ΔA^*, ΔE^*, and ΔS^*); e.g., ΔA and ΔA^* are estimated from

$$\Delta A = A^{unfolded} - A^{folded} = -k_B T \ln(\frac{P_u}{P_f}), \quad (7)$$

$$\Delta A^* = A^{TS} - A^{unfolded} = -k_B T \ln(\frac{P_{TS}}{P_u}), \quad (8)$$

where P_f , P_u and P_{TS} are the probabilities of the folded, unfolded, and transition states, respectively. Note that the optimized $\lambda(INC,NC,E)$ in Table 5.2 are only suitable to estimate p_B values for configurations constrained in the phase space between the two stable states, i.e., configurations with $\lambda \leq 0$ and $\lambda \geq 1$ values are assigned to the p_B=0 and p_B=1 ensembles, respectively. P_{TS} was obtained by encompassing the free energy maximum close to the ensemble of configurations with p_B=0.5. The energy change associated with Eq. (7) [or (8)] can be computed from the difference between the average configurational energy of the folded state [or TS] and unfolded state. The entropic contributions can be estimated from $T\Delta S = \Delta E - \Delta A$ and $T\Delta S^* = \Delta E^* - \Delta A^*$. For all the split systems, the changes in A, E, and S as the system goes from the unfolded to the TS and to folded states are depicted in Figure 5.5.

The reason why we calculate differences with respect to the folded state in Eq. (7) is because the energy and entropy (and thus the free energy) of the "unique" folded configuration is approximately the same for all systems (e.g., $E^{folded} \sim -20.5 k_B T$, $S^{folded} \approx 0$); consequently, comparing ΔA, ΔE, and ΔS across systems highlights the differences in the unfolded states. While such differences in unfolded states is likely very large in configuration space, Figure 5.5 shows that they are rather small in the A, E, and S spaces ($<kT$), once the unfolded state is standardized (or "renormalized") according to our p_B=0 model definition; indeed, all split system show a similar transitional behavior. Another general feature in Figure 5.5 is that the entropy change between the TS and the folded state is minimal for all systems, indicating that most of the translational and conformational freedom has been lost by the time the TS is reached (recall that $S^{folded} \approx$

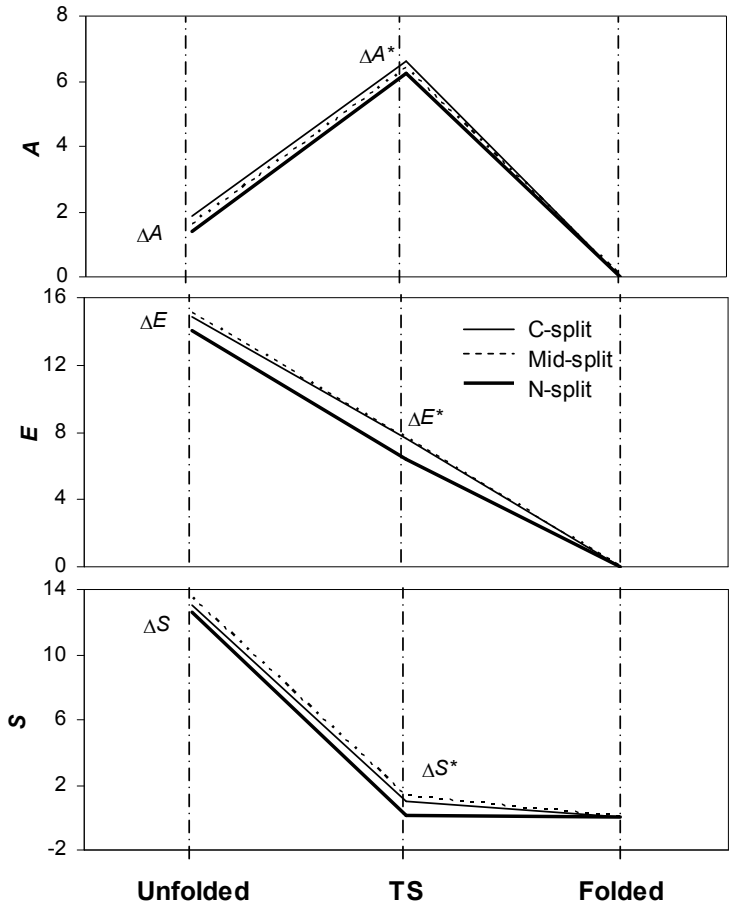

Figure 5.5 Schematic showing the difference in free energy, energy, and entropy for the different split systems as they go from the unfolded state, through the transition state (TS), and to the folded or reassembled state. For convenience the properties for the folded state are all set to the same value of zero.

0). In the context of transition state theory where the rate constant is $k_{A\to B} \approx C \exp(-\Delta A^*/kT)$, the similarity of ΔA^* for all split systems suggests that thermodynamic quantities alone will not be strongly indicative of kinetic behavior, and that differences in the "frequency factor" C are likely significant (keeping in mind that transition state theory itself has a limited interpretative value as suggested by Figure 5.3). Nonetheless, some useful correlations can be established in light of the small but real differences in energetic and entropic changes.

In the N-split case, the unfolded state is characterized by conformations having no interactions between the two fragments, open conformations of the small chain A, and compact, prefolded conformations for chain B. The C-split system exhibits a more cooperative folding behavior where the unfolded state is characterized by configurations with both chains attached such that the total entropy of the system is essentially purely conformational. These characteristics of the unfolded state cause a nontrivial interplay of the interactions between the fragments. For example, given that the concentration of the folding core amino acids in a single fragment (chain B) stabilizes the unfolded state, the N-split system has a stronger (more negative) energetic interactions than the unfolded C-split case ($E_{N-split}^{unfolded} - E_{C-split}^{unfolded} = -0.67kT$); resulting in a smaller enthalpic driving force in going from unfolded to the folded state ($\Delta E_{N-split} < \Delta E_{C-split}$ in Figure 5.5). In contrast, the total entropy drop (penalty) from the unfolded to the folded state is smaller, more favorable for the N-split; i.e., $\Delta S_{N-split} < \Delta S_{C-split}$. Although the drop in translational entropy is expected to the larger for the N-split case, the prefolding of the large fragment at early stages of the folding decreases its configurational entropy, resulting in a smaller overall entropy drop than for the other split systems (as shown in Figure 5.5). Overall, the free energy drop from the unfolded to the folded state for the N-split is slightly smaller than for the other systems ($\Delta A_{N-split} < \Delta A_{M-split}^{uf} < \Delta A_{C-split}^{uf}$ in Figure 5.5), indicating that the

reassembly driving force is smaller for the N-split system. In the following, differences among systems are discussed in the context of the kinetic FFS data.

The mechanistic details for the reassembly transition of the split systems were obtained by collecting ensembles of configurations at different iso-lines of the committor surface p_B = 0.2, 0.5, and 0.8 (shown in Figures 5.3 and 5.4). The configurations were classified during a FFS simulation using $\lambda=p_B$ model 2 (in Table 5.2) as reaction coordinate. Each such ensemble was then analyzed at a microscopic level by determining the probability of each native contact pair to belong to the corresponding ensemble.

For the N-split system, Figure 5.6 shows the contact density map and snapshots of typical configurations for the p_B = 0.2, 0.5, and 0.8 ensembles. Native interactions between the two fragments begin at late stages of the folding process as the TSE is characterized by structures in which interchain interactions constitute only 10% (2 out of 20) of the most probable native contacts. Interestingly, 69% and 63% of those native contacts with a minimum 80% probability to belong to the p_B=0.2 and p_B=0.5 surfaces, respectively, correspond to the unsplit-system CC set. The nucleus formation starts in the chain B at early stages of the folding process (p_B=0.2) by forming 73% of the CC set. Additionally, 80% of those most probable native contacts at the p_B =0.2 and 0.5 surfaces are formed in the chain B which indicates that this chain pre-folds, separate from the chain A which remains as a random coil. It is only after the reaction pathway crosses the p_B=0.5 transition state (TS) dividing region, that the number of native contact between chains and intrachain contacts in chain B increase to achieve the final folded state. However, only 2 out of 30 of those most probable native contacts at the p_B=0.8 iso-surface correspond to intrachain contacts in the small A fragment, indicating that the chain A completes its folding at a very late stage of the reassembly process. This kinetic analysis fully supports the picture we

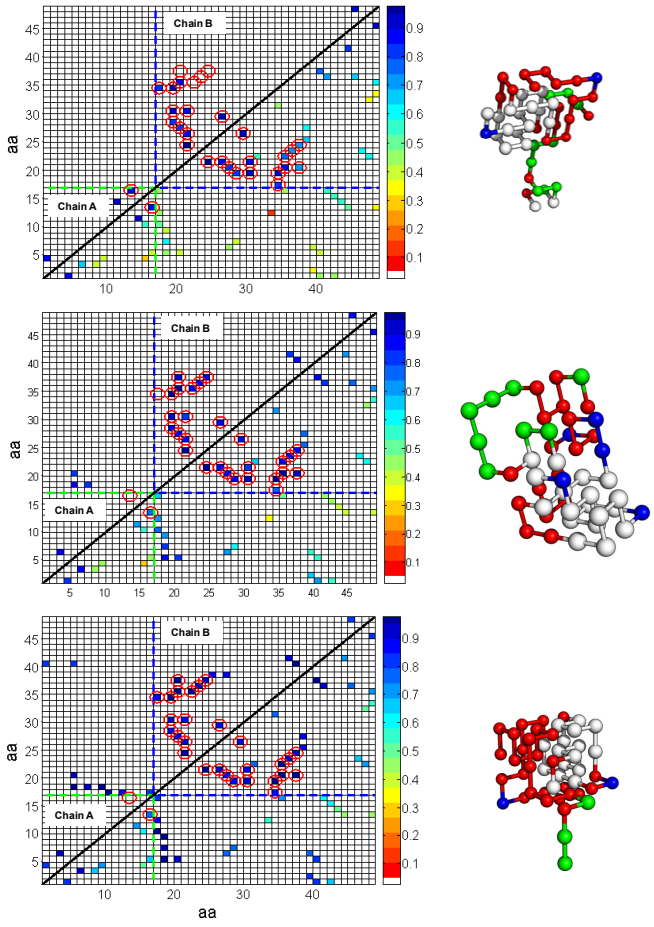

Figure 5.6 Contact density map for the N-split system for the ensembles of configurations belonging to isocommittor surfaces: p_B=0.2 (top), p_B=0.5 (center), and p_B=0.8 (bottom). The x- and y-axis represent the amino acid (aa) position in the 48-mer sequence. The ensembles were collected by estimating p_B values for all the interfacial points in the TPE from reaction coordinate model 2 in Table II. The lower triangle (below the diagonal line) shows the probability of a native contact to belong to the ensemble. The upper triangle shows those native contacts with at least 80% probability to belong to the corresponding p_B ensemble. Encircled symbols represent native contacts that form the critical folding nucleus. Snapshots depicting typical configurations observed for each ensemble are also shown where green indicates chain A, blue indicates chain B, red indicates intrachain native contacts, and white indicates native contacts that form the critical folding nuclei. The large squares in dashed lines enclose native intrachain contacts: (green) chain A and (blue) chain B.

gathered before from a thermodynamic analysis (18) which argued that while the prefolding of the chain B favors folding on entropic grounds (i.e., the unfolded and folded states have similar amount of order) it disfavors folding on energetic grounds [since the unfolded state is stabilized at lower energies closer to the folded state energy as seen in Figure 5.2 (b)]. For the N-split system, this interplay of the interactions causes retardation of the folding mechanism.

In the case of the fast folding C-split system, native interactions between chains starts at early stages of the folding process such that interchain associations constitute ~20 and 30% of the most probable native contacts at the p_B=0.2 and p_B=0.5 isosurfaces, respectively. Figure 5.7 shows the contact density map and snapshots of typical configurations for the p_B = 0.2, 0.5, and 0.8 surfaces. CC contacts are only 31% and 35% of those native contacts with at least 70% probability to belong to the p_B=0.2 and p_B=0.5 ensembles, respectively. Moreover, the same 10 most probable CC contacts formed at the TSE are also observed in the p_B=0.8 ensemble. Hence, we can infer that the nucleation in the C-split system uses a different set of native CC compared to the 48-mer protein, i.e., a different folding nucleus. This is inconsistent with our previous conjecture (18) that the C-split and the 48-mer systems had a similar nucleation mechanism because they had the TS dividing surface located in the same position along the conformational energy as reaction coordinate [see Figure 3 in Ref. (18)].

Furthermore, in our previous work (18) we hypothesized that the fact that the TS dividing region is shifted to the folded basin evidenced that the reassembly of the N-split system takes place via a different, slower folding mechanism compared to the Mid- and C-split systems [see Figure 3 at Ref. (18)]. However, our results in Figure 5.6 suggest that the N-split folding mechanism entails a fast nucleation event in the large fragment (chain B) which is significantly similar to that of the folding nucleus

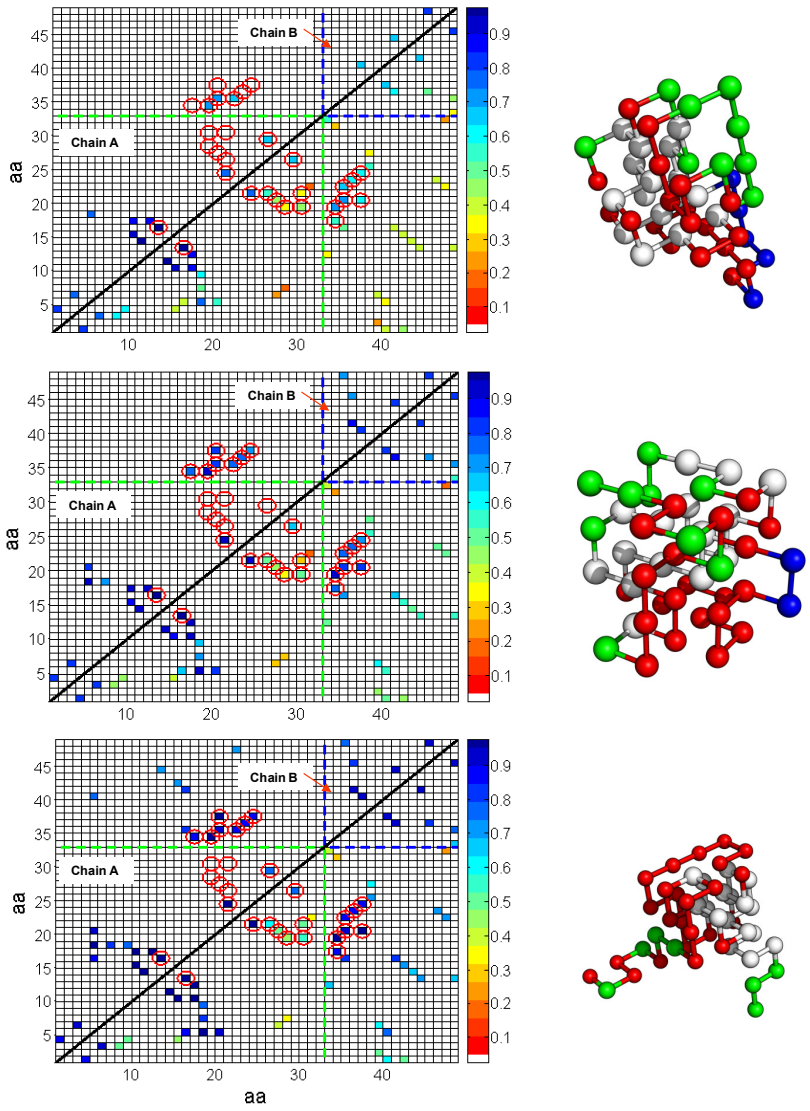

Figure 5.7 Contact density map for the C-split system for ensembles of configurations belonging to isocommittor surfaces: p_B=0.2 (top), p_B=0.5 (center), and p_B=0.8 (bottom). The axes, symbols, lines, and colors have the same interpretation as given in the caption of Fig. 6, except that the upper triangle shows those native contacts with at least 70% probability to belong to the given p_B ensemble.

(CC set) in the unsplit system. Of course that the similarity in the core contacts at the TS surface for the N-split and unsplit systems, evidencing a similarity in the nucleation process, need not translate into a faster reassembly process for the N-split system relative to the other split systems. These results for the C-split and N-split systems illustrate that any conclusions drawn based on shifts (or lack thereof) of the TS dividing surface are risky, especially if a poor order parameter is used as reaction coordinate as was the case with the conformational energy in Ref. (18). Figure 5.7 also shows that the set of most probable native contacts at the TSE includes 6 out of 7 intrachain contacts in chain B which suggests that this small chain is folded at that stage. Such a prefolding leads to a reduction of the total entropy for the system at the TS, facilitating the transition to the folded state.

For the Mid-split system, the contact map density and snapshots of typical configurations for the p_B = 0.2, 0.5, and 0.8 surface regions are shown in Figure 5.8. Similar to the C-split system, the native interactions between chains begin at early stages of the folding process. Hence, interchain associations correspond to ~56 and 67% of those native contacts with at least 70% to belong to the ensemble of configurations at p_B = 0.2 and TSE, respectively. However, there are two main differences between the folding mechanisms for the Mid- and C-split systems; in the former case the CC set provides a shared folding nucleus between chains that "glues" fragments together, and the folding of both chains is preceded by a cooperative association of the chains (i.e., they fold while attached one to another). Figure 5.8 suggests that the transition of the Mid-split system to the folded state follows a nucleation mechanism similar to that observed in the unsplit system. Indeed, 67% and 93% of the CC set is formed by those native contacts most likely to occur in the p_B = 0.2 and p_B = 0.5 ensembles, respectively. In this case, a prefolded state of one of the chains is not observed at the TSE (in contrast to the C-split case), indicating that the

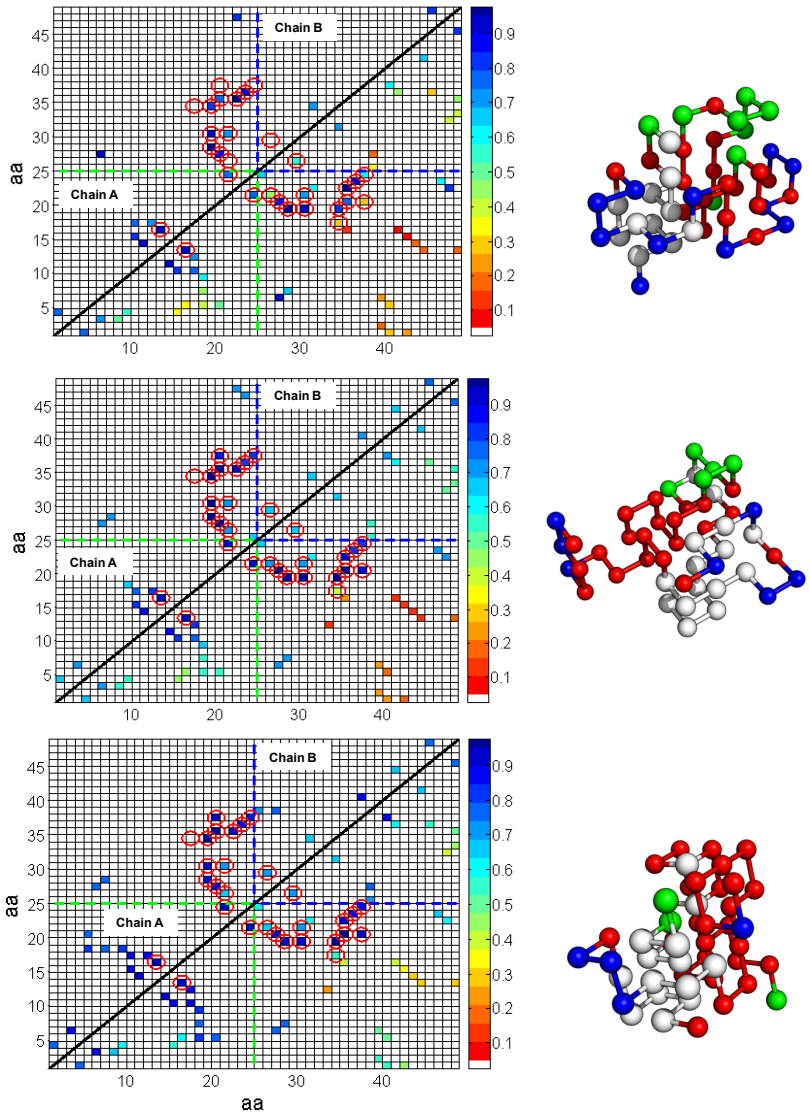

Figure 5.8 Contact density map for the Mid-split system for the ensembles of configurations belonging to isocommittor surfaces: p_B=0.2 (top), p_B=0.5 (center), and p_B=0.8 (bottom). The axes, symbols, lines, and colors have the same interpretation as given in the caption of Fig. 6, except that the upper triangle shows those native contacts with at least 70% probability to belong to the given p_B ensemble.

formation of intrachain contacts in both chains is given by a cooperative folding behavior. As observed in Figure 5.4, the average rate constant for the Mid-split transition is nearly 2 times larger than that of N-split transition, whereas the C-split folding is 2 times faster than the Mid-split transition. Note that the early prefolding of the small chain for the C-split decreases slightly its configurational entropy compared to the Mid-split case ($\Delta S_{C-split} < \Delta S_{Mid-split}$ in Figure 5.5). The early strong interchain association for the Mid-split system balances out the early prefolding of the small chain in the C-split system to produce comparable energy changes [e.g., $\Delta E_{C-split} \approx \Delta E_{Mid-split}$) ; overall, the free-energy drop is slightly larger for the C-split system.

A common characteristic shared by the faster folding systems (Mid- and C-split) is that the formation of a few CC native contacts at an early stage facilitates the reassembly via a more cooperative behavior. We then hypothesized that if a single point mutation is introduced to the N-split case, resulting in a new "MN-split" system, by strengthening one of the prevalent native interchain contacts observed in the TSE, then chains would associate earlier (i.e., at p_B=0.2) and accelerate reassembly. To test this idea, the MN-split case incorporated a "pseudo point-mutation" that had a twice larger-than-normal contact energy for residue 5 in chain A and residue 2 in chain B (this was the stronger interchain NC pair observed in the TSE for the N-split system as seen in Figure 5.6). Figure 5.9 shows the corresponding contact map density and snapshots of typical configurations for the p_B = 0.2, 0.5, and 0.8 isosurfaces. As expected, the pseudo mutation causes chain interactions at early stages of the folding process, where interchains associations correspond to 25% and 40% of those native contacts with at least 80% probability to belong to the p_B = 0.2 and p_B = 0.5 ensembles, respectively. Moreover, Figure 5.9 also suggests that the transition of the NM-split system to the folded state still follows a nucleation mechanism similar to that observed

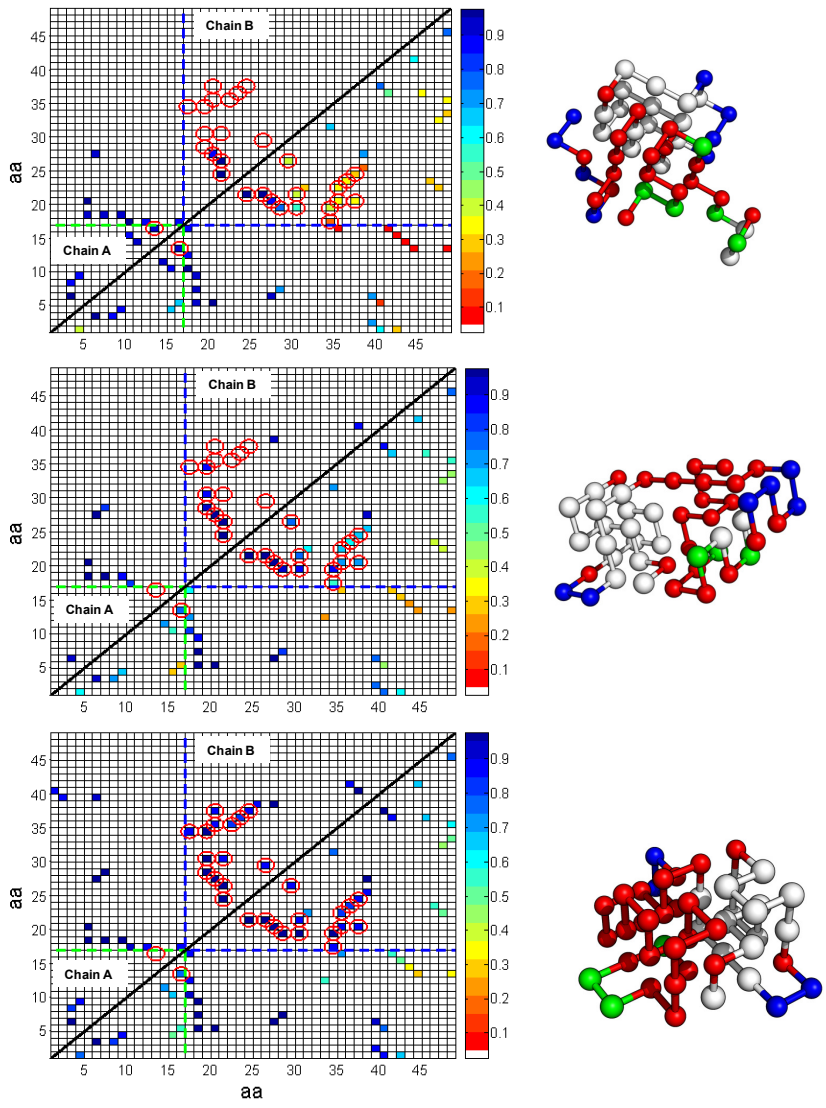

Figure 5.9 Contact density map for MN-split system for the ensembles of configurations belonging to isocommittor surfaces: p_B=0.2 (top), p_B=0.5 (center), and p_B=0.8 (bottom). The axes, symbols, lines, and colors have the same interpretation as given in the caption of Fig. 6, except that the upper triangle shows those native contacts with at least 70% probability to belong to the given p_B ensemble.

in the N-split system, since now 27%, 40% and 100% of the CC set is formed by those most probable native contacts in the p_B = 0.2, 0.5, and 0.8 ensembles, respectively. Since 50% and 78% of those most probable native contacts at the p_B =0.5 and 0.8 surfaces, respectively, are formed in the chain B (in which the CC set is concentrated), indicates that this chain pre-folds. Figure 5.4 confirms that the pseudo point-mutation speeds up reassembly compared to the unmutated N-split case. In particular, the average rate constant for the MN-split transition is ~2 times greater than that of the N-split transition.

VI. Supplementary Material

A. Optimization of interface positioning and sampling for initial order parameter λ = NC.

The number of native contacts was used as initial guess of the order parameter (i.e., λ=*NC*). Accordingly, we defined λ_0 as the upper limit of the unfolded state region (i.e., $\lambda_0 = \lambda_A$), by taking $\lambda_A \le 9$, while the final region *B* is the folded state with $\lambda_B = \lambda_n = 58$. We conducted a preliminary BG run to first optimize the $\{\lambda_i\}$ set (see Sec. IIC). We used initially n=12 interfaces to partition the phase space, as listed in Table 5.4 in the supplementary material. The initial number of trials per point at λ_i was $k_0 = 100$ and $k_i = 10$ ($1 \le i < n$) and $P(\lambda_{i+1} \mid \lambda_i)$ values were averaged over ten blocks. Each block consisted of 100 BG runs starting from a randomly selected configuration at λ_0 (out of the N_0 points collected therein) from which a branched path is generated. We chose $\langle p_B \rangle_{\lambda_i} = \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.8\}$ for $5 \le \lambda_i \le n$-1 and $P(\lambda_{i+1} \mid \lambda_i) = P_\lambda$ for $0 \le \lambda_i \le 4$, where $P_\lambda = \left[P(\lambda_{n=B} \mid \lambda_0) / \prod_{i=1}^{n-1} P(\lambda_{i+1} \mid \lambda_i)\right]^{1/5}$ = constant; the latter ensures that at least 5 interfaces are placed in the region close to the basin of attraction A (i.e., in the steeply uphill region of $\langle p_B \rangle_\lambda \le 0.1$). In this way, the λ interfaces are distributed over the entire phase space region: the region close to basin B ($11 \le \lambda_i \le 12$), the uphill

Table 5.4 Optimized $\{\lambda'\}$ and $\{k'\}$ sets for the initial guess of the order parameter λ=NC. The $\langle p_B \rangle$ values were approximated by $\langle p_B \rangle^{\lambda_{n-1}} = P(\lambda_{n=B} \mid \lambda_{n-1})$ for i=n-1 and $\langle p_B \rangle^{\lambda_{n-i}} = P(\lambda_{n-i+1} \mid \lambda_{n-i}) \langle p_B \rangle^{\lambda_{n-i+1}}$ for $0 \le i \le n$-2.

i	Initial $\{\lambda\}$ set	*Optimized* $\{\lambda'\}$ and $\{k'\}$ set				*Optimized* $\{\lambda'\}$ and $\{k'\}$ set				*Optimized* $\{\lambda'\}$ and $\{k'\}$ set			
	λ_i	C-split				Mid-split				N-split			
		λ_i'	k_i'	$P(\lambda_{i+1}\|\lambda_i)$	$\langle p_B \rangle$	λ_i'	k_i'	$P(\lambda_{i+1}\|\lambda_i)$	$\langle p_B \rangle$	λ_i'	k_i'	$P(\lambda_{i+1}\|\lambda_i)$	$\langle p_B \rangle$
0	9	9	100	0.130	~0.000	9	100	0.113	~0.000	9	100	0.064	~0.000
1	10	12	6	0.069	~0.000	12	6	0.098	~0.000	13	6	0.074	~0.000
2	11	17	10	0.076	~0.000	16	8	0.117	~0.000	18	8	0.096	~0.000
3	12	22	12	0.113	~0.000	20	8	0.196	0.002	25	8	0.247	0.001
4	14	27	12	0.106	0.005	23	9	0.106	0.008	32	9	0.087	0.006
5	16	31	11	0.208	0.050	28	7	0.339	0.077	37	10	0.189	0.067
6	18	33	5	0.745	0.237	31	4	0.709	0.228	38	5	0.815	0.354
7	20	35	6	0.720	0.319	33	5	0.854	0.321	39	6	0.866	0.434
8	23	37	7	0.738	0.443	34	6	0.773	0.377	40	7	0.899	0.501
9	26	39	8	0.828	0.600	36	7	0.816	0.487	41	8	0.831	0.558
10	29	41	8	0.835	0.726	38	8	0.736	0.597	43	9	0.866	0.671
11	33	44	7	0.869	0.869	44	8	0.811	0.811	46	10	0.775	0.775
12 (=n)	58	58	----	----	----	58	----	----	----	58	----	----	----

region close to basin A ($0 \le \lambda_i \le 5$ for $\langle p_B \rangle_\lambda \le 0.1$), and the transition state (TS) region ($8 \le \lambda_i \le 10$ for $0.40 \le \langle p_B \rangle_\lambda \le 0.60$. The $\{k_i\}$ set was also optimized after having already chosen a suitable λ staging; the new $\{k_i'\}$ set was found by letting k_0'= 100 and the other k_i' values to follow Eq. (31) in Ref. (19); which are given in Table 5.4. However, we set k_i=10 to be the minimum acceptable value needed to obtain p_B estimates with accuracy at the second significant digit.

B. Graphical correlation of p_B with significant and non-significant variables

Supplementary Figures 5.10-5.12 show average values for all the candidate collective variables on the ensemble of configurations belonging to the isosurfaces p_B = 0.1, 0.2, 0.3, 0.5, 0.8, and 1.0. These ensembles were collected by estimating p_B values for all interfacial points from model 2 during a FFS simulation. The averages <INC>, < NC> and <E> values increases monotonically, showing an one-to-one correspondence between an isocomittor p_B surface region and the average on the ensemble of configurations. However, for the non-significant variables (i.e., <NNA>, <NNB>, <CC>, and <IC>) determined from ANOVA, a same ensemble average value could corresponds to different isocomittor p_B surface regions which indicates that this variable is unable to describe the progress of the folding transition. Hence, according to the ANOVA analysis, INC, NC and E variables are macroscopic properties of the system that capture the intrinsic folding transition of the unsplit systems.

VII. Outlook

While path sampling methods have a widespread application to many types of rare transitional events in biological systems, including isomerizations, protein folding, and enzyme catalysis. It is important to keep in mind, however, that the successful implementation of novel path sampling methods (and of forward flux

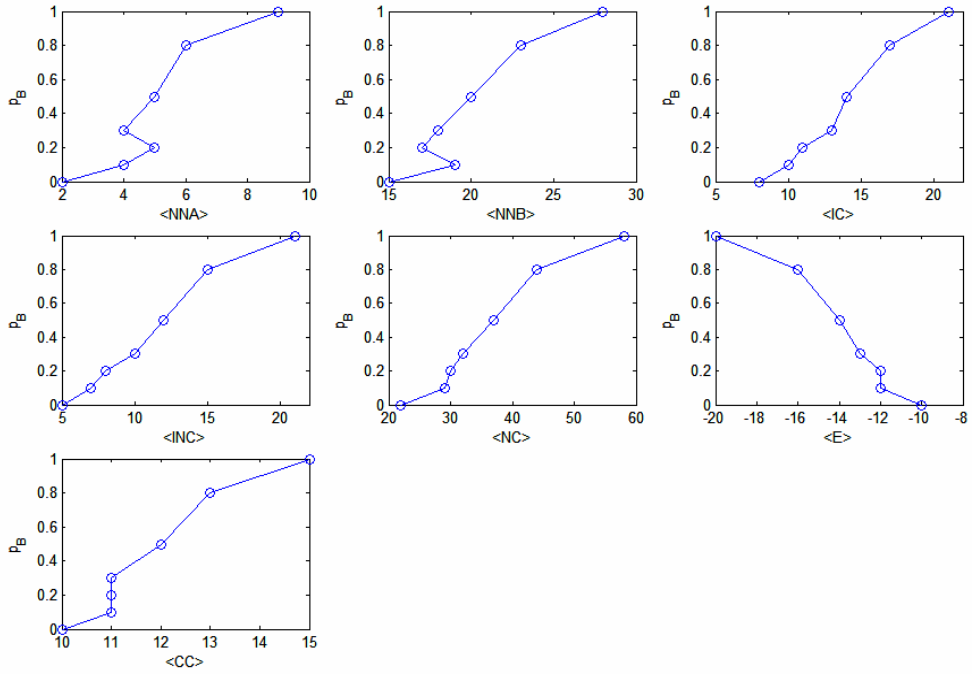

Figure 5.10 Averages for the set of candidates collective variables (see Sec. III D) as function of the isocommittor p_B surfaces for the N-split system.

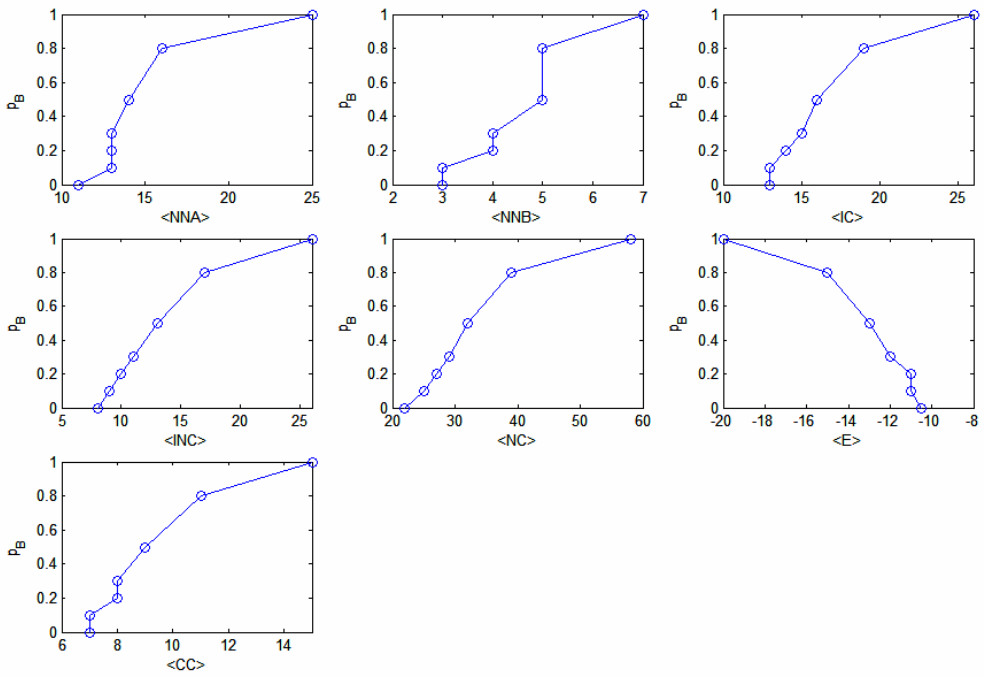

Figure 5.11 Averages for the set of candidates collective variables (see Sec. III D) as function of the isocommittor p_B surfaces for the C-split system.

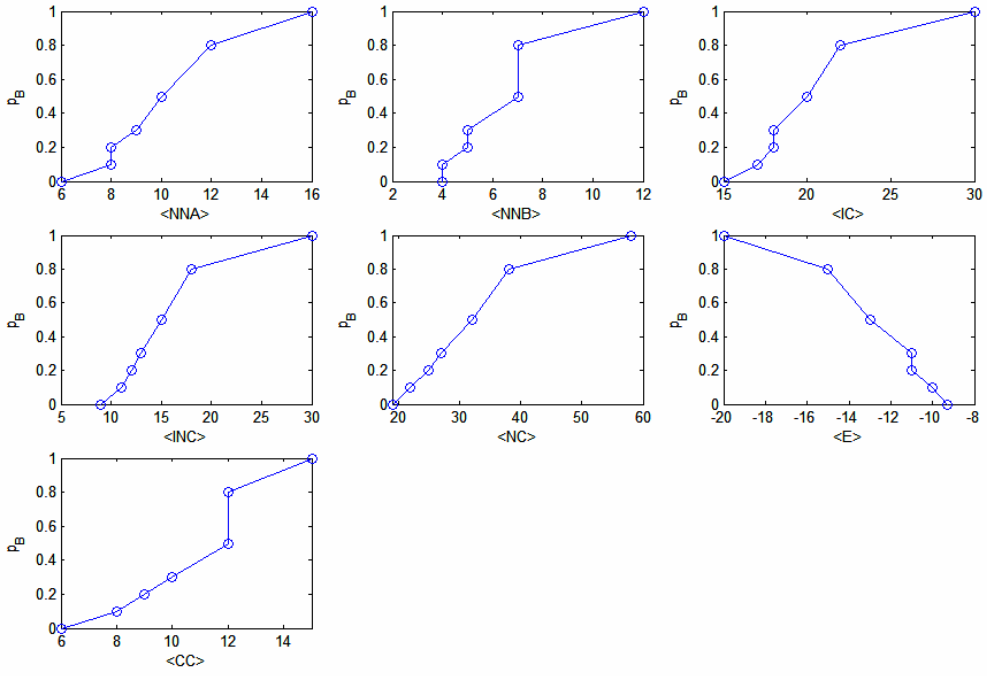

Figure 5.12 Averages for the set of candidates collective variables (see Sec. III D) as function of the isocommittor p_B surfaces for the Mid-split system.

sampling in particular) crucially depends on the careful selection of the method's parameters; this selection should not be based only on experience or "art" but also on systematic optimization strategies such as those discussed in this work. Hence, the proposed optimized forward flux sampling methods could become a valuable tool for unraveling new mechanistic details of important biomolecular processes, including folding of a biomolecule into its "native" state, protein-protein signaling, protein-antigen binding, DNA-enzyme interactions during replication, protein misfolding and aggregation, biomolecular isomerization, genetic switches, and biopolymer translocation and motion through pores and membranes.

However, optimization approaches that combine forward flux sampling simulations with methods employing accelerated dynamics and other time-saving strategies (like coarse graining) will be needed to study the countless biological systems that not only entail very many degrees of freedom, but also exhibit a multiplicity of intermediates and transition channels, and span very broad time scales.

REFERENCES

1. Allen, R. J., D. Frenkel, and P. R. ten Wolde. 2006. Forward flux sampling-type schemes for simulating rare events: Efficiency analysis. Journal of Chemical Physics 124:194111.

2. Allen, R. J., D. Frenkel, and P. R. ten Wolde. 2006. Simulating rare events in equilibrium or nonequilibrium stochastic systems. Journal of Chemical Physics 124:024102.

3. Allen, R. J., P. B. Warren, and P. R. ten Wolde. 2005. Sampling rare switching events in biochemical networks. Physical Review Letters 94:018104.

4. Bolhuis, P. G. 2003. Transition path sampling on diffusive barriers. Journal of Physics: Condensed Matter 15:S113-S120.

5. Bolhuis, P. G., D. Chandler, C. Dellago, and P. L. Geissler. 2002. Transition path sampling: Throwing ropes over rough mountain passes, in the dark. Annual Review of Physical Chemistry 53:291-318.

6. Dellago, C., P. G. Bolhuis, F. S. Csajka, and D. Chandler. 1998. Transition path sampling and the calculation of rate constants. Journal of Chemical Physics 108:1964-1977.

7. Dellago, C., P. G. Bolhuis, and P. L. Geissler. 2002. Transition path sampling. Advances in Chemical Physics 123:1-78.

8. Faradjian, A. K., and R. Elber. 2004. Computing time scales from reaction coordinates by milestoning. Journal of Chemical Physics 120:10880-10889.

9. Moroni, D., T. S. van Erp, and P. G. Bolhuis. 2004. Investigating rare events by transition interface sampling. Physica a-Statistical Mechanics and Its Applications 340:395-401.

10. van Erp, T. S., and P. G. Bolhuis. 2005. Elaborating transition interface sampling methods. Journal of Computational Physics 205:157-181.

11. van Erp, T. S., D. Moroni, and P. G. Bolhuis. 2003. A novel path sampling method for the calculation of rate constants. Journal of Chemical Physics 118:7762-7774.

12. Dellago, C., and P. G. Bolhuis. 2007. Transition path sampling simulations of biological systems. In Atomistic Approaches in Modern Biology: from Quantum Chemistry to Molecular Simulations. Springer-Verlag Berlin, Berlin. 291-317.

13. Dellago, C., and P. G. Bolhuis. 2008. Transition path sampling and other advanced simulation techniques for rare events. In Advanced Computer Simulation Approaches for Soft Matter Sciences III. Eds. C. Holmer, K. Kremer, Springer. 167-233.

14. Borrero, E. E., and F. A. Escobedo. 2006. Folding kinetics of a lattice protein via a forward flux sampling approach. Journal of Chemical Physics 125:164904-164914.

15. Fersht, A. R. 1995. Optimization of Rates of Protein-Folding - the Nucleation-Condensation Mechanism and Its Implications. Proceedings of the National Academy of Sciences of the United States of America 92:10869-10873.

16. Fersht, A. R. 1997. Nucleation mechanisms in protein folding. Current Opinion in Structural Biology 7:3-9.

17. Borrero, E. E., and F. A. Escobedo. 2007. Reaction coordinates and transition pathways of rare events via forward flux sampling. Journal of Chemical Physics 127:164101-164117.

18. Martinez, L. M. C., E. E. B. Quintana, F. A. Escobedo, and M. P. DeLisa. 2008. In silico protein fragmentation reveals the importance of critical nuclei on domain reassembly. Biophysical Journal 94:1575-1588.

19. Borrero, E. E., and F. A. Escobedo. 2008. Optimizing the sampling and staging for simulations of rare events via forward flux sampling schemes. Journal of Chemical Physics 129:024115-024116.

20. Frenkel, D., and B. Smit. 2002. Understanding Molecular Simulation: From Algorithms to Applications. Academic, Boston.